

GALOIS SWITCHING FUNCTIONS :

Algebraic Structures and Applications

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

by

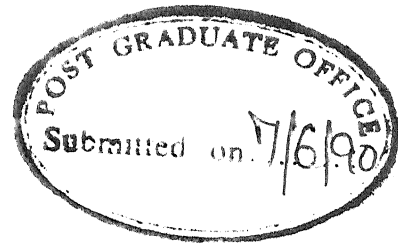
GEORGE VARGHESE

to the

DEPARTMENT OF ELECTRICAL ENGINEERING


INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JUNE, 1990



CERTIFICATE

It is certified that the work contained in the thesis entitled 'GALOIS SWITCH FUNCTIONS: Algebraic Structures and Applications', by Mr. George Varghese, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.


(M. U. SIDDIQI) 7.6.90

Professor

Department of Electrical Engineering

Indian Institute of Technology

Kanpur.

June, 1990

SYNOPSIS

George Varghese
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
India

GALOIS SWITCHING FUNCTIONS : Algebraic Structures and Applications

This thesis is concerned with algebraic structures and applications of G switching functions (GSFs). GSFs are a generalization of binary switching functions (Boolean functions) with domain and range assuming values from finite (Galois) fields $GF(p^k)$ and $GF(p^n)$ respectively, where p is a prime and integer k is not necessarily less than or equal to integer n . The treatment of GSFs in this thesis is confined to the practically important case of $p = 2$. Although GSFs are of interest in a wide range of areas such as switching systems, error control coding, cryptography and image processing, applications of GSFs considered in this thesis are restricted to the areas of characterization, classification and synthesis of switching functions, and error control coding.

Switching functions over finite fields have been studied by several authors. However, only a few results are available on algebraic structures and properties of these functions. This aspect of GSFs is emphasized in this thesis and properties of signals representable by GSFs are studied in an algebraic framework.

Advantages of spectral characterization of discrete signals and systems defined over finite index sets are well known. Specifically, discrete Fourier transform (DFT) over finite fields has been employed extensively in error control coding for characterization of signals. However, the utility of DFT is restricted to those signal lengths that are relatively prime to the field size.

27 AUG 1992
CENTRAL LIBRARY
I. I. T. KANPUR
Acc. No. A.1.14020

7h
621.381537

V 426 G

EE-1880-D-VAR-GAL

SYNOPSIS

George Varghese
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
India

GALOIS SWITCHING FUNCTIONS : Algebraic Structures and Applications

This thesis is concerned with algebraic structures and applications of Galois switching functions (GSFs). GSFs are a generalization of binary switching functions (Boolean functions) with domain and range assuming values from finite (Galois) field $GF(p^k)$ and $GF(p^n)$ respectively, where p is a prime and integer k is not necessarily equal to integer n . The treatment of GSFs in this thesis is confined to the practically important case of $p = 2$. Although GSFs are of interest in a wide range of areas such as switching systems, error control coding, cryptography and image processing, applications of GSFs considered in this thesis are restricted to the areas of characterization, classification and synthesis of switching functions, and error control coding.

Switching functions over finite fields have been studied by several authors. However, only few results are available on algebraic structures and properties of these functions. The algebraic aspect of GSFs is emphasized in this thesis and properties of signals representable by GSFs are studied in an algebraic framework.

Advantages of spectral characterization of discrete signals and systems defined over finite index sets are well known. Specifically, discrete Fourier transform (DFT) over finite fields has been employed extensively in error control coding for characterization of codes. However, the utility of DFT is restricted to those signal lengths that are relatively prime

the characteristic of the finite field. One solution to this problem is to impose alternative structures on the index set of signals under consideration so that a finite field transform which can accommodate signal lengths that are not relatively prime to the characteristic of the field, can be defined on them. One such structure is that of a cyclic monoid. The algebra of discrete signals whose index set has the structure of a cyclic monoid is called a *cyclic monoid algebra*.

GSFs qualify to be members of a multiplicative cyclic monoid algebra $M(2^k)$ of dimension 2^k . The nonzero elements of index sets of GSFs constitute a multiplicative cyclic group of order $2^k - 1$. However, the multiplicative inverse of the '0' element of the index set is not defined, thus giving rise to the structure of cyclic monoids to the index sets. The two binary operations in the cyclic monoid algebra are pointwise addition and an appropriately defined convolution. The cyclic monoid algebra is isomorphic to a residue class polynomial algebra over an appropriate finite field extension. The two binary operations in this algebra are polynomial addition and polynomial multiplication modulo $(x^{2^k} - x)$. The isomorphism between these two algebras is a finite field transform, called Galois Transform (GT), which transforms convolution in the function domain to pointwise multiplication in the spectral domain. This transform is essentially an extension of DFT over finite fields, thus making it possible for conjugacy relations in the case of the latter to be extended to the former. Polynomials representing GSFs under this isomorphism (transform) are called Galois polynomials (GPs). It follows that the coefficients of the GP representing a GSF are the GT coefficients of the signal vector over $GF(2^n)$ of length 2^k . If the transform vectors lie in an extension field of $GF(2^n)$, then the GP representing a GSF is shown to have remarkable properties which provide a means for their realization through parallel processing techniques. This is because conjugacy relations permit the terms in the GP to be grouped into disjoint Frobenius cycles which can be realized independently. Since computation of Frobenius sum in a Frobenius cycle involves repeated squaring of an element in that cycle

the efficiency of computation can be increased by exploiting normal basis (NB) representation of finite field elements, as squaring of an element represented in NB amounts to mere cyclic shift of the components of its cartesian representation with respect to that basis.

A class of functions where the theory of GSFs can be immediately applied is the class of k -variable Boolean functions (BFs), since they are a subclass of the general class of GSFs, where the mapping is from $GF(2^k)$ to $GF(2)$. It is shown that any k -variable BF has a Frobenius polynomial (FP) representation, ie., its GP coefficients satisfy conjugacy constraints, allowing the terms in its GP representation to be grouped into Frobenius cycles. A study of the BFs as members of a monoid algebra over $GF(2)$ is carried out, and standard classes of BFs like the linear and β -self dual (SD)/anti self dual (ASD) BFs are characterized in the transform domain. Spectral domain study of linear Boolean functions (LBFs) reveals the fact that they are ideals in monoid algebras over $GF(2)$. This result is then applied to the class of generalized Reed-Muller (GRM) codes to show that they can also be described by ideal structures in appropriate monoid algebras over $GF(2)$, as they are constructed from LBFs.

Although transform domain studies on β -SD/ASD BFs do not show any specific algebraic structures in monoid algebra, the transform coefficients are shown to satisfy certain constraints which help in their identification. Characterization of β -SD/ASD BFs for 2, 3 and 4 variables is carried out in terms of their GP coefficients and constraints on the coefficients are derived.

Classification problem of BFs is considered and the existing equivalence relations for classification of BFs (commonly known as the five invariance operations) and their effect on the coefficients of GPs representing BFs are studied, as a consequence of which a class identification procedure for 2 and 3 variable BFs by verification of their GP coefficients is formulated and a finite field model which synthesizes any BF of a class from a prototype function of that class is proposed. Alternately, an attempt is made to see whether certain

operations connected with the monoid algebra model of BFs would be suitable for classification purposes of the same. The case of 2 and 3 variable BFs is examined and it is shown that each of these classes have members from different ideals in appropriate monoid algebras. A finite field model for BF synthesis, which sums up elements from ideals, is proposed. This is a Frobenius sum computer which can make use of NB for implementation purposes.

Traditional transform domain characterization of linear block codes adopts the practice of representing individual code vectors in the spectral domain by finite field DFT techniques. But, as pointed out earlier, this is not possible for all code lengths since DFT of all lengths does not exist in the finite field case. A possible way to overcome this limitation is to regard linear block codes as mappings from the k -tuple vector space to the n -tuple vector space and view them as signals over multiplicative cyclic monoids $M(2^k)$ which assume values from a finite field $GF(2^n)$, thus making all linear block codes amenable to transform domain studies, and allowing them to be characterized by single variable GPs over an appropriate extension field. It is shown that any linear (n,k) block code is a linearized GSF (LGSF) which represents a one-to-one mapping, and that the linearized Galois polynomial (LGP) representing this mapping has, in general, k coefficients belonging to $GF(2^L)$, where $L = \text{L.C.M. of } n \text{ and } k$. These LGPs representing linear mappings constitute a subclass of the general class of GPs. Depending on whether conjugacy relations among the LGP coefficients are nontrivial or trivial, the corresponding polynomials are called linearized Frobenius polynomials (LFPs) and linearized polynomials (LPs) respectively.

The fact that any one-to-one linear mapping is representable by a LGSF admits the possibility of any general linear mapping, which is not necessarily one-to-one, also to be represented by a LGSF. Thus a study of the general class of LGSFs which also includes those which represent linear block codes as a subclass, is taken up and an isomorphism between linear (n,k) transformations (linear transformations from the k -tuple vector space

to the n -tuple vector space where k is not necessarily equal to n) and LGSFs represented by LGPs over $GF(2^L)$, is established. It is further shown that the class of LGSFs constitutes an ideal in the corresponding monoid algebra over $GF(2^n)$, out of which the subclass of LGSFs representing one-to-one mappings (and hence linear (n,k) block codes) have LGP representations whose coefficients satisfy certain nonzero determinant property. Classes of LGSFs are studied in terms of the nature of the mappings generated by them.

The algebra of LGPs is studied with specific reference to linear block codes in terms of an operation of composition known as symbolic multiplication of LGPs. The class of LGSFs represented by single term LGPs is examined in detail. It is shown that any single term LGP representing a linear (n,k) transformation, where k divides n , always represents a one-to-one mapping and hence a linear (n,k) block code. Groups of single term LGPs are shown to have the structure of finite fields isomorphic to $GF(2^n)$ with the operations of addition and symbolic multiplication.

Coefficients of a LGP representing a linear block code are obtained from the basis vectors of the code. Consequently, there are as many LGP representations of a linear block code as the number of ways a basis can be chosen for the same. Thus given two LGPs which are known to represent linear block codes, it would be desirable to know whether they represent the same code or different codes. Results in this direction are achieved for codes generated by single term LGPs which are members of the finite fields mentioned earlier. A study of distinctness of the codes in these fields is conducted and the number of distinct codes in each field is computed. It is shown that when n and k are relatively prime, all the codes in the respective finite field are distinct.

A study of the roots of LGPs representing linear (n,k) block codes is conducted next. It is observed that the roots of LGPs need not necessarily belong to the same field. Further, it is shown that they characterize groups of codes rather than individual codes. It is also shown that they cannot assume nonzero values from $GF(2^k)$.

Standard basis and normal basis LGP representations of cyclic codes are derived

from a canonic form of their basis vectors. It is shown that for some (n,k) cyclic codes whose k divides n , the NB LGP representation is simply a q -polynomial over $GF(q)$, ie., a LGP with coefficients from the ground field (if the ground field under consideration is $GF(p)$, then the LGP is denoted as p -polynomial).

Standard array decoding problem of linear block codes is considered. Since the standard array is essentially a two-dimensional (2-D) truth table, it has been possible to compactly represent standard arrays using 2-D GSFs, on lines similar to those for representing the usual one-dimensional (1-D) truth tables as 1-D GSFs. It is shown that a wide variety of options are open for both 1-D and 2-D GSF implementation of standard array decoders depending on whether the received vector is to be decoded into a k -tuple message vector or an n -tuple code vector. It is shown that any 1-D GSF which maps the received vector into a k -tuple message can be implemented by a Frobenius sum computer and hence NB representations and parallel processing techniques can be used to advantage in such situations for fast decoding of linear block codes. GP representation of syndromes is considered and it is pointed out that any syndrome table has a linearized Frobenius polynomial (LFP) representation, ie., a LGP whose coefficients satisfy conjugacy constraints. In general, the roots of these polynomials belong to an extension field of $GF(2^n)$. However, those roots which lie in $GF(2^n)$ are shown to be the code vectors of the corresponding linear block code.

Possibility of characterizing linear block codes by the roots of appropriate LGP leads to an alternate characterization of linear block codes by means of syndromic polynomials (SPs). A SP is a LP of degree 2^k over $GF(2^n)$ whose roots are non-repetitive and constitute the code vectors of a linear (n,k) block code. They represent special types of LGSFs characterizing many-to-one linear mappings from $GF(2^n)$ to $GF(2^n)$ of particular kind: These mappings are such that any element of $GF(2^n)$ which is a member of a given k -dimensional subspace of $GF(2^n)$ gets mapped into the '0' element of $GF(2^n)$ whereas any other element which is not a member of that subspace gets mapped into

member of an $(n-k)$ dimensional subspace of $GF(2^n)$ which constitutes the root space of another SP called its dual polynomial which also represents a similar mapping. In other words, every SP of degree 2^k has associated with it a k -dimensional subspace as its root space and an $(n-k)$ dimensional subspace as its range space which constitutes the root space of a dual SP of degree 2^{n-k} , and vice versa. Except the fact that the roots of a LP have the structure of a subspace, very few results are available on these polynomials (SPs) as far as characterization of the subspaces are considered. An investigation of properties of SPs is carried out and it has yielded fruitful results. It is shown that any LP in x of degree 2^k over $GF(2^n)$ with the coefficient of x nonzero, and which divides $x^{2^n} - x$, uniquely characterizes a linear (n,k) block code; the code vectors being the roots of the polynomial. Using this property of LPs and the associated duals, it is proved that SPs can be used for decoding of linear block codes which they represent as root spaces. It is shown that they in fact can be used for computation of syndromes of the respective codes, thus accounting for the name SP, the syndromes being members from the root space of the dual SP. Thus these syndromes are n -tuples in contrast to the syndromes usually associated with a standard array, which are $(n-k)$ tuples.

Reference to representations of finite field elements with respect to NB so far has been from the point of view of certain implementational advantages. Their role in the characterization and study of linear block codes using SPs has turned out to be even more significant. SPs representing linear block codes whose code vectors are considered as elements with respect to some NB of $GF(2^n)$ have been called normal basis syndrome polynomials (NB SPs). Such representations are noteworthy because of the following facts :

First, it is shown that it is possible to identify codes of the same weight distribution from their NB SPs. Secondly, NB SP representation helps in the characterization of t -cyclic codes (quasi-cyclic codes which are closed under t cyclic shifts, $t \geq 1$). Specifically, it is shown that any linear (n,k) t -cyclic code has a NB SP representation whose

coefficients belong to a subfield $GF(2^t)$ of $GF(2^n)$, and conversely, any SP with coefficients from $GF(2^t)$, which divides $x^{2^n} - x$, represents a t -cyclic code.

The third fact which goes in favour of NB representations follows from the second. For $t = 1$, we get the characterization of the important class of cyclic codes which has a NB SP representation in the form of a p -polynomial. A different proof of this result on cyclic codes is also given to emphasize the fact that a cyclic subspace has the structure of a modulus when represented with respect to a NB. The dual of the SP of any linear block code, in general, does not represent the SP of the corresponding dual code. However, in the case of cyclic codes represented in NB, the dual polynomial and the SP representing the dual cyclic code are shown to be the same. This follows from the theory of p -polynomials where conventional polynomial arithmetic and symbolic arithmetic are related through the notion of q -associates. The NB p -polynomial representation of a cyclic code is easily derivable from the generator polynomial of its dual cyclic code and is shown to be equal to the linearized q -associate of the same.

The final point in favour of NB representations is a new approach to the study of weight distribution of cyclic codes. This is based on factorizing their NB p -polynomial representations. An algorithm for factorizing polynomials over finite fields, based on DFT over finite fields, is developed. This algorithm is particularly efficient if there are no repetitive roots and if the field in which the roots lie are known; both of these conditions are satisfied by a SP. It is shown that the number of cycles in a cyclic code is equal to the number of irreducible polynomials in the factorization of its NB p -polynomial, the number of members in each cycle is equal to the degree of each irreducible polynomial in the factorization, and a cycle representative of the code is given by a representative root of each irreducible polynomial. Examples of NB p -polynomial representations of Bose—Chaudhuri—Hocquenghem (BCH) and Golay codes are given and their weight distributions are studied by factorization of respective NB p -polynomial representations.

Self dual (n,k) cyclic codes are characterized in terms of their NB p -polynomial representations and it is shown that these polynomials split in $GF(2^k)$, as a consequence of which the study of their weight distributions reduces to finding the number of Frobenius classes in $GF(2^k)$, the order of each Frobenius class, and the weight of a representative member of each class expressed in NB cartesian form in $GF(2^n)$. This respectively gives the number of cycles in the code, number of members in each cycle, and information about its weight distribution.

ACKNOWLEDGMENTS

Words are not enough to express my sincere and heartfelt gratitude to my thesis supervisor Dr. M.U. Siddiqi for his valuable guidance and constant encouragement, without which this thesis would never have shaped up. Even in his busy hours, he was always patient enough to help me out of the problems I faced in my research work. Besides, he gave me complete freedom to work in the Image Processing Lab. Thank you very much, sir.

I wish to express my thanks to all my teachers in this department who have taught me various courses and updated my knowledge.

The association with my colleagues Udaya and Madhu who were always ready to assist me in my research problems can never be forgotten. Thank you, friends.

I express a special word of thanks to Venkatesh and Deepak Murthy for drawing those neat diagrams in this thesis and especially to the latter for his assistance during the final preparation of the thesis. I am thankful to my lab mates Govinda Rajan, Ramprasad, Hariprasad, Subramanyam and others for the company and help they have given me.

I thank Prof. P.O.J. Lebba, Principal of my college, the college authorities and the Director of Technical Education, Kerala, for sponsoring me for this programme and the Q.I.P. cell for providing financial support.

I remember the love and affection of my wife Jessy and daughter Renu and the patience with which they spent all these years of my study period. Thank you, dears.

Last but not least, I thank Lord Jesus for the blessings he has bestowed upon me throughout my life.

Not by my merits

But by His Grace

Dedicated to

my Lord Jesus Christ

who has given me the wisdom

to undertake this work and has cared for

me in all my needs

TABLE OF CONTENTS

	Page
LIST OF TABLES	xxi
LIST OF FIGURES	xxv
LIST OF SYMBOLS AND ABBREVIATIONS	xxvi
 CHAPTER 1 INTRODUCTION	 1
1.1 Scope of the Work	1
1.2 Historical Background	3
1.3 Outline of Chapters	6
 CHAPTER 2 THEORY OF GALOIS SWITCHING FUNCTIONS	 12
2.1 Notion of a Galois Switching Function	13
2.2 Algebraic Models for Galois Switching Functions (GSFs)	16
2.2.1 Monoid Algebra Model of GSFs	16
2.2.2 Transform Domain Description of GSFs	19
2.3 Galois Polynomial Representation of GSFs	22
2.3.1 Residue Class Polynomial Algebra Model of GSFs	25
2.3.2 Modified Cyclic Monoid Algebra Model for GSFs	26
2.3.3 Takahashi's Representation of GSFs	29
2.4 Frobenius Cycles in GPs	30
2.4.1 Conjugacy Relations	31
(a) $k \nmid n$	31
(b) $k \mid n$	33
	34

3.7	Relating the Coefficients of a Linearized GP to the Vectors Generating the Corresponding Linear (n,k) Transformation	60
3.8	Conjugacy Relations in Linearized GPs	65
(a)	$k \nmid n$	66
(b)	$k \mid n$	72
3.9	Algebraic Structures of Single Term Linearized GPs	77
3.9.1	Group Structure of GPs of the form $\rho^j f(x)$, $j = -\infty, 0, \dots, 2^n - 2$.	77
3.9.2	Algebraic Structure of Single Term Linearized Polynomials	78
3.9.3	Algebraic Structure of Single Term Linearized Frobenius Polynomials	79
3.9.3.1	Frobenius Symbolic Multiplication	79
3.9.3.2	Finite Field Structure	81
CHAPTER 4	GSF THEORY FOR BOOLEAN FUNCTIONS	87
4.1	Representation of Boolean Functions by GPs	87
4.2	Monoid Algebra Model of Boolean Functions	88
4.2.1	Ideals in the Monoid Algebra of Boolean Functions	89
4.3	Algebraic Characterization of Linear Boolean Functions	91
4.3.1	Representation of Linear Boolean Functions by Linearized GPs	91
4.3.2	Linear Boolean Functions as Ideals in a Monoid Algebra	92
4.4	Algebraic Characterization of Generalized Reed-Muller (GRM) Codes	93
4.4.1	Representation of the Basis Vectors of Binary GRM Codes by GPs	95

4.4.2	GRM Codes as Ideals in a Monoid Algebra	96
4.5	Classification of Boolean Functions	100
4.5.1	The Five Invariance Operations	101
4.5.1.1	Invariance Operations on the Domain of Boolean Functions	102
4.5.1.2	Invariance Operations on the Range of Boolean Functions	103
4.5.1.3	Combining the Operations	103
4.5.2	Effect of the Five Invariance Operations on the GP Coefficients	104
4.5.3	Class Identification by Verifying the GP Coefficients	108
4.5.4	Operations Based on the Monoid Algebra Structure of Boolean Functions	110
4.5.4.1	Convolution Operation with a Function whose GP Coefficients are $a_{-w} = 0, a_i = \gamma^{-i}, i = 0, 1, \dots, 2^k - 2.$	111
4.5.4.2	Convolution Operation on Arbitrary Boolean Functions	114
4.6	Finite Field Models for Boolean Function Synthesis	115
4.6.1	Model Based on the Five Invariance Operations	116
4.6.2	Model Based on Frobenius Sum Computation	117
4.7	Characterization of β -Self Dual (SD)/Anti Self Dual (ASD) Boolean Functions by GPs	119
4.7.1	Derivation Strategy	120
4.7.2	Characterization of 2-Variable β -SD/ASD Boolean Functions	122
(1)	β -Self Dual Functions	122
(2)	β -Anti Self Dual Functions	124
4.7.3	Characterization of 3-Variable β -SD/ASD Boolean Functions	126

(1)	β -Self Dual Functions	126
(2)	β -Anti Self Dual Functions	128
4.7.4	Characterization of 4-Variable β -SD/ASD Boolean Functions	130
(1)	β -Self Dual Functions	130
(2)	β -Anti Self Dual Functions	135
CHAPTER 5	GSF THEORY FOR ERROR CONTROL CODES	139
5.1	Representation of Linear Codes by Linearized GPs	141
5.1.1	Number of Linearized GPs Representing Linear Codes	141
5.2	Condition for Linearized GPs to Represent Linear Codes	142
5.3	Representation of Classes of Linear Codes of the Same Weight Distribution by Linearized GPs	146
5.4	Nature of Linear Mappings Generated by Linearized GPs of the form $\beta^j f(x)$, $j = 0, 1, \dots, 2^n - 2$	150
5.5	Nature of Linear Mappings Generated by Single Term Linearized GPs	152
5.6	Nature of the Linear Codes Generated by Single Term Linearized GPs which are Members of a Finite Field	152
5.6.1	Linear Codes Generated by Single Term LPs of F_l	153
5.6.2	Linear Codes Generated by Single Term Linearized Frobenius Polynomials of F_f	159
5.7	Roots of Linearized GPs Representing Linear Codes	169
5.8	Representation of Cyclic Codes by Linearized GPs	169

5.8.1	Standard Basis Representation	171
5.8.2	Normal Basis Representation	172
5.9	Decoding of Linear Codes Using GSFs	179
5.9.1	The Standard Array Principle	179
5.9.2	Representation of Standard Array Using Two-Variable GPs	180
5.9.3	Standard Array Decoding Using GSFs	183
(i)	Using 1-D GSFs	183
(a)	Decoding into an n -tuple Code Vector	184
(b)	Decoding into a k -tuple Message Vector	186
(ii)	Using 2-D GSFs	189
(a)	Decoding into an n -tuple Code Vector	190
(b)	Decoding into a k -tuple Message Vector	192
5.9.4	Syndrome Tables and their Representation Using GPs	194
CHAPTER 6	SYNDROME POLYNOMIAL REPRESENTATIONS OF LINEAR BLOCK CODES	196
6.1	Representation of a Linear Code as the Root Space of a Linearized Polynomial	197
6.2	Linearized Polynomials for Decoding of Linear Codes	198
6.3	Normal Basis Syndrome Polynomials	201
6.4	Normal Basis Syndrome Polynomial Representations of Linear Codes with the Same Weight Distribution	202
6.5	Normal Basis Syndrome Polynomial Representations of Quasi Cyclic Codes	207
6.6	Normal Basis Syndrome Polynomial Representations of Cyclic Codes	213
6.6.1	Representation of a Cyclic Code as the Root Space of a Normal Basis P-Polynomial	213

6.6.2	Computation of the Normal Basis P-Polynomials Representing a Given (n,k) Cyclic Code and its Dual (n,n-k) Cyclic Code	215
6.7	Study of Weight Distributions in Cyclic Codes	219
6.7.1	Determination of the Weight Distribution of Cyclic Codes from their Normal Basis P-Polynomial Representations	220
6.7.2	Examples of Normal Basis P-Polynomial Representations of BCH Codes	224
6.7.3	Examples of Normal Basis P-Polynomial Representations of Golay Codes	230
6.7.4	Normal Basis P-Polynomial Representations of Self Dual Cyclic Codes	231
CHAPTER 7	CONCLUSIONS	232
7.1	Summary of Results	233
7.2	Suggestions for Further Work	240
APPENDIX A	MATHEMATICAL BACKGROUND	242
A.1	Basic Algebraic Structures	242
A.2	Discrete Fourier Transform (DFT) over Finite Fields	265
A.3	Linearized Polynomials	267
APPENDIX B	FACTORIZATION OF POLYNOMIALS OVER FINITE FIELDS USING DFT OVER FINITE FIELDS	280
APPENDIX C	TABLES OF FINITE FIELDS	285
REFERENCES		288

LIST OF TABLES

Table No.		Page
2.1	Truth Table Representation of a System of 2 Boolean Functions of 3 Variables	13
2.2	Truth Table of the System of Boolean Functions of Table 2.1 Considered as a Mapping from $GF(2^3)$ to $GF(2^2)$ with the Input Variables in Natural Order	15
2.3	Truth Table of the Mapping Given in Table 2.2 with the Input Variables in Field Order	15
3.1	Cayley Tables for the Finite Field F_f Comprising of Single Term Linearized Frobenius Polynomials Representing Linear (3,2) Codes	86
(a)	Frobenius Symbolic Multiplication Table	86
(b)	Addition Table	86
4.1	Minimal Ideals in a Monoid Algebra Consisting of k-Variable Boolean Functions	90
(a)	$k = 2$	90
(b)	$k = 3$	90
4.2	GP Coefficients of the Code Vectors of a First Order GRM Code of Block Length 8 Considered in Example 4.4.1	100
4.3	Classification of k-Variable Boolean Functions using Convolution as Defined in Monoid Algebra with a Function whose GP Coefficients are $a_{-\infty} = 0, a_i = \gamma^{-i}, i = 0, 1, \dots, 2^k - 2.$	112
(a)	$k = 2$	112
(b)	$k = 3$	112
4.4	Classification of 3-Variable Boolean Functions using Convolution as Defined in Monoid Algebra on Arbitrary Functions	115

4.5	2-Variable β -Self Dual Boolean Functions and their GP Representations	123
(a)	$\beta = 1$	123
(b)	$\beta = \gamma$	124
(c)	$\beta = \gamma^2$ (self dual)	124
4.6	2-Variable β -Anti Self Dual Boolean Functions and their GP Representations	125
(a)	$\beta = 1$	125
(b)	$\beta = \gamma$	125
(c)	$\beta = \gamma^2$ (anti self dual)	125
4.7	3-Variable Self Dual Boolean Functions and their GP Representations	128
4.8	3-Variable Anti Self Dual Boolean Functions and their GP Representations	130
4.9	4-Variable Self Dual Boolean Functions and their GP Representations	133
4.10	4-Variable Anti Self Dual Boolean Functions and their GP Representations	136
5.1	Nonzero Members of F_7 Comprising of Single Term LPs Representing Linear (6,3) Codes	156
5.2	First Block of nonzero members of a Group Comprising of Multiple Term LPs Representing the Same Set of Codes as in the First Block of F_7 in Table 5.1	159
5.3	Single Term Linearized Frobenius Polynomials Representing One-to-One Linear (3,2) Transformations Grouped into 6 Isomorphic Finite Fields	163
5.4	First Block of Nonzero Members of F_f comprising of Single Term Linearized Frobenius Polynomials Representing Distinct Linear (6,4) Codes	166
5.5	First Block of Nonzero Members of a Group Comprising of Multiple Term Linearized Frobenius Polynomials Representing the Same Set of Distinct Linear Codes	168

	as in Table 5.4	
5.6	Representation of the Standard Array of a Linear (4,2) Code Using Two-Variable GPs	182
(a)	Standard Array for the Code	182
(b)	Row Transform of the Standard Array in (a)	182
(c)	Column Transform of the Matrix in (b)	182
5.7	Decoding Table for the Linear (4,2) Code considered in Example 5.9.1 with the Received n -tuples as Domain and the Transmitted n -tuples as Range	185
5.8	Decoding of a Linear (4,2) Code into k -tuple Message Vectors using 1-D GSFs	187
(a)	Standard Array for the Code	187
(b)	Decoding Table for the Code with the Received n -tuples as Domain and transmitted k -tuples as Range	187
5.9	Standard Array for the Linear (5,2) Code of Example 5.9.4	189
5.10	Decoding of a Linear (4,2) Code into n -tuple Code Vectors Using 2-D GSFs	191
(a)	Decoding Table with the Received n -tuples (split into k -tuples and $n-k$ tuples) as Domain and the Transmitted n -tuples as Range	191
(b)	Coefficients of the Two-Variable GP Representing (a)	191
5.11	Decoding of a Linear (5,2) Code into k -tuple Message Vectors Using 2-D GSFs	193
(a)	Decoding Table with the Received n -tuples (split into k -tuples and $n-k$ tuples) as Domain and the Transmitted k -tuples as Range	193
(b)	Coefficients of the Two-Variable GP Representing (a)	193
6.1	Standard Array for a Linear (5,2) Code	200
6.2	Standard Array of Table 6.1 including Syndromes (expressed in Polar Form)	201

6.3	Normal Basis Syndrome Polynomials of Linear (4,2) Codes grouped into Classes on the basis of Same Weight Distributions	205
6.4	Normal Basis Syndrome Polynomials of (6,3) 2-Cyclic Codes grouped into Classes on the basis of Same Weight Distributions	210
6.5	Normal Basis Syndrome Polynomials of (6,3) 3-Cyclic Codes grouped into Classes on the basis of Same Weight Distributions	211
6.6	Representation of a (7,3) Cyclic Code in Normal Basis	217
6.7	Representation of a (7,4) Cyclic Code (which is the Dual of the (7,3) Cyclic Code given in Table 6.6) in Normal Basis	218
6.8	Irreducible Polynomials in the Factorization of $x^{63} + x^7 + 1$ and their Representative Roots in Normal Basis	223
6.9	Irreducible Polynomials in the Factorization of $G(x)/x$ and their Representative Roots in Normal Basis	226
6.10	Irreducible Polynomials in the Factorization of $H(x)/x$ and their Representative Roots in Normal Basis	226
6.11	Irreducible Polynomials in the Factorization of $G(x)/x$ and their Representative Roots in Normal Basis	228
A.1	List of Primitive Polynomials over $GF(2)$ of degree n ; $2 \leq n \leq 15$	251
A.2	Number of Different Normal Bases in $GF(2^n)$; $2 \leq n \leq 15$	264
C.1	$GF(2^2)$	285
C.2	$GF(2^3)$	285
C.3	$GF(2^4)$	286
C.4	$GF(2^5)$	287

LIST OF FIGURES

Table No.

- 2.1 Horner's Polynomial Computer
- 4.1 Finite Field Models for Boolean Function
Synthesis
 - (a) Model Based on the Five Invariance
Operations
 - (b) Model Based on Frobenius Sum Computation

LIST OF SYMBOLS AND ABBREVIATIONS

(x)	:	symbolic multiplication
$*$:	convolution as defined in monoid algebra
\otimes	:	cyclic convolution
\otimes	:	direct product
\oplus	:	direct sum
\ominus	:	modulo subtraction
$+$:	modulo 2 addition or ex—or
\cup	:	inclusive or
\in	:	belongs to
\notin	:	does/do not belong to
\equiv	:	congruent to
\longleftrightarrow	:	transform pairs
$k n$:	k divides n
$k \nmid n$:	k does not divide n
$GF(.)$:	Galois field or finite field
$\text{frs}(\Theta)$:	Frobenius sum of an element Θ
$\text{tr}(\Theta)$:	trace of an element Θ
Z_2	:	field of 2 elements
$GL_k(Z_2)$:	The set of all invertible linear transformations acting on a k -dimensional vector space over the field Z_2 , called the <i>General linear group</i>
$A_k(Z_2)$:	The set of all affine transformations acting on a k -dimensional vector space over the field Z_2 , called the <i>Affine group</i>

\bar{x}_1	:	complement of x_1
\bar{x}, x^\dagger	:	complementing all the k -variables in x
iff	:	if and only if
Q.E.D.	:	end of proof
L.C.M.	:	least common multiple
G.C.D.	:	greatest common divisor
RAG	:	restricted affine group
DFT	:	discrete Fourier transform
GSF	:	Galois switching function
LGSF	:	linearized Galois switching function
BF	:	Boolean function
LBF	:	linear Boolean function
FF	:	Frobenius function
LFF	:	linearized Frobenius function
LF	:	linearized function
GP	:	Galois polynomial
FP	:	Frobenius polynomial
LGP	:	linearized Galois polynomial
LFP	:	linearized Frobenius polynomial
LP	:	linearized polynomial
SP	:	syndrome polynomial
NB	:	normal basis
SB	:	standard basis

CHAPTER 1

INTRODUCTION

1.1 Scope of the Work

This thesis gives the results of a study of Galois switching functions (GSFs) with regard to their algebraic structures and applications. GSFs are a generalization of binary switching functions (Boolean functions) where the input and output variables belong to finite fields of characteristic p . The treatment in this thesis is confined to $p = 2$. Earlier, studies of GSFs [1–11] have concentrated on topics such as finding closed form expressions, minimization techniques and computational advantages. Here, GSFs are considered in an algebraic framework. For this purpose, the algebraic models proposed in [12] are employed, where GSFs are considered as discrete signals over finite index sets with the structure of multiplicative cyclic monoids $M(2^k)$. Accordingly, the study of GSFs is essentially viewed as a study of cyclic monoid algebras over finite fields.

An advantage resulting from attributing the structure of a cyclic monoid to the domain values of GSFs is that it is then possible to conduct transform domain studies on signal lengths that are not relatively prime to the characteristic of the finite field. Further, the transform domain study results in compact representations of discrete signals defined over multiplicative cyclic monoids which include switching functions and error control codes, as a consequence of which alternative structures for their realization becomes possible. In these structures, the basic building blocks would consist of multi output modules based on finite field arithmetic suitable for polynomial computations. It is hoped that such realizations may lead to better systems in terms of chip count and computation time. Further, the so-called conjugacy relations among the coefficients of Galois

polynomials (GPs) representing GSFs provide a natural means for their realization and processing through parallel processing techniques.

Application areas considered in this thesis have been confined to those of switching functions and error control codes, even though there is considerable scope for their utility in other areas like cryptography, image processing and fault tolerant computing.

GSFs corresponding to the following mappings have been investigated in detail:

- (1) General mappings from $GF(2^k)$ to $GF(2^n)$, with particular reference to specific mappings from $GF(2^k)$ to $GF(2)$ resulting in Boolean functions (BFs) and their characterizations.
- (2) General linear mappings from $GF(2^k)$ to $GF(2^n)$, where k is not necessarily equal to n and the mapping not necessarily one-to-one, representing linear (n,k) transformations, described by a special class of GPs, called linearized Galois polynomials (LGPs); corresponding GSFs are called linearized GSFs (LGSFs). The LGPs are called linearized Frobenius polynomials (LFPs), if their coefficients satisfy nontrivial conjugacy relations and called simply linearized polynomials (LPs) if the conjugacy relations are trivial. The associated LGSFs are respectively called linearized Frobenius functions (LFFs) and linearized functions (LFs).
- (3) Specific linear mappings from $GF(2^k)$ to $GF(2^n)$, where $k < n$, and the mapping is one-to-one, giving rise to linear (n,k) block codes, and their representations by LFPs and LPs as the case may be.
- (4) Specialized linear mappings from $GF(2^n)$ to $GF(2^n)$ which are many-to-one and described by a pair of LPs of degree 2^k and 2^{n-k} respectively, with the k -dimensional root space of the former constituting the range space of the latter and vice versa, both spaces being subspaces of $GF(2^n)$. These mappings provide alternate characterizations of linear (n,k) block codes. Their linearized polynomial (LP) representations are called syndrome polynomials (SPs).

1.2 Historical Background

Boolean algebra has been conventionally used for switching function analysis and synthesis. In Boolean algebra, given a truth table of a 'k' input, 'n' output digital system, a set of 'n' Boolean functions (BFs) $f_j(x_0, x_1, x_2, \dots, x_{k-1})$, $j = 0, \dots, n-1$, of 'k' variables over GF(2) is constructed. We get 'n' Boolean sums-of-products (SOP) expressions which may be minimized using Karnaugh map or Quine-McClusky procedure to give networks with fewer AND and OR gates. Use of Galois fields has been suggested as an alternative to Boolean algebra [10]. Because then any given truth table can be represented by a one-dimensional (1-D) GSF over an extension field of GF(2), in place of 'n' functions of 'k' variables as in Boolean algebra.

Study of switching functions based on Galois fields was initiated by Ninomiya [1, 2]. Use of extension fields for this purpose was considered by Bartee and Schneider [3] and Benjauthrit and Reed [4, 5]. Menger, Jr. [6] pointed out the discrete Fourier transform (DFT) relationship between the coefficients and the function values of a GP representing a GSF, if the domain values are represented as a power of a primitive element in the corresponding extension field. Further, he proposed the use of multi output modules for BF synthesis which are capable of handling arithmetic in finite field extensions, in place of conventional two-state systems. The finite field modules which perform addition and multiplication in finite field extensions have been called respectively as PLUS and TIMES modules by Menger and he has suggested the possible use of VLSI technology for their fabrication. Minimization of GSF expressions to give networks with fewer modules has also been considered by him. He has proposed a minimization technique based on factorization of polynomials. Minimization problems in GSFs have also been taken up by Pradhan and Patel [7] besides Mukhopadhyay and Schmitz [8]. Their ideas are based on Reed-Muller codes. Pradhan has also formulated a theory of GSFs in [9].

Authors mentioned in the above paragraph have considered switching functions where the input, output and the function values belong to the same finite field. Takahas

[10] has considered switching functions whose input-output pairs are not restricted to the same field, because of which, the coefficients of the GPs representing these switching functions are forced to be chosen from a larger field which contains both the input and output fields as subfields. In other words, in Takahashi's representation, the coefficients of GPs belong to a field whose extension order is equal to the least common multiple (L.C.M.) of the extension orders of the input and output fields. By this, he has been able to exploit the DFT relations between the function values and the coefficients to their fullest extent.

Literature on the algebraic structures of GSFs is scant. Davio, Deschamp and Thayse [11] have discussed linear algebra structure of GSFs. Recently, Siddiqi and Sinha [12] have formulated isomorphic algebraic models for GSFs. In one model, GSFs are considered as members of a *cyclic monoid algebra* over a finite field, with the two binary operations being pointwise addition and an appropriately defined convolution. An isomorphic algebraic model which they have described is a *residue class polynomial algebra* model. In this model, 1-D GSFs are described by single variable polynomials with appropriate polynomial arithmetic defined on them. In this thesis, we show that these models allow us to view Takahashi's polynomial representation of GSFs in an algebraic framework. Frobenius properties of GPs are regarded as a consequence of the monoid algebraic structure of GSFs because of which an appropriate DFT-like transform exhibiting conjugacy relations could be defined on these functions.

Problems encountered in switching theory have been dealt with by several authors [13, 14, 15, 16, 17] and spectral techniques over the field of real numbers have been suggested to tackle them. Certain transforms like Hadamard, Paley-Walsh (P-W) [15], and Rademacher-Walsh (R-W) [13, 14, 17] transforms have been used as tools for transforming Boolean data into the spectral domain. Use of the five invariance operations for classification of BFs based on R-W transform is given in [13, 14, 17] where classification of BFs has been carried out using them for 2, 3, 4 and 5 variable cases. The

number of classes are respectively found to be 2, 3, 8 and 48. The five invariance operations are so called because they do not change the magnitude of the R-W coefficients. These operations collectively constitute a group known as the *restricted affine group (RAG)* as observed in [16].

Finite field spectral domain study of linear block codes has appeared in literature in the form of Mattson-Solomon (M-S) polynomials [18]. Generalized Reed-Muller (GRM) codes and their M-S polynomial representations are given in [19]. Blahut [20, 21] has used DFT over finite fields to study codes in the spectral domain. He has also discussed GRM codes from a spectral point of view using the notions of radix q -weight of integers suggested by Kasami, Lin and Peterson [22].

Origin of LPs can be traced back to the fundamental papers of Ore [23, 24, 25, 26] which contain their theory in detail, including the study of LPs under the operation of symbolic multiplication. Lidl and Niederreiter, in their book [27] give an extensive bibliography on LPs, besides discussing their theory. Other books on this topic are by Berlekamp [28] and MacWilliams & Sloane [19]. Theory of LPs followed in this thesis is mainly on the lines of Lidl & Niederreiter [27] with occasional references to Berlekamp [28] and MacWilliams & Sloane [19].

LPs have been associated with linear transformations when the output vectors in the transformation belong to a finite field which is either same as, or a subfield of the input field, the associated polynomial being a trace function in the latter case [27]. To our knowledge, association of a linearized Galois polynomial (LGP) over $GF(2^L)$, where $L = \text{L.C.M. of } n \text{ and } k$, with general linear (n, k) transformations, where k is not necessarily equal to n , has not been considered in literature. We have conducted a study in this direction.

As far as syndrome polynomials (SPs) are concerned, most of the text books on finite fields and coding theory [19, 27, 28, 29] discuss these polynomials for associating them with k -dimensional subspaces and dual polynomials, but the properties of these

polynomials which help in characterizing linear block codes have not been dealt with. A reference to these polynomials as *root polynomials* is given by Jamison [30], who applies them to the problem of covering vector spaces with cosets of subspaces. We have utilized these polynomials for characterization, decoding and weight distribution studies of linear block codes in general and cyclic codes in particular. Further, we have shown that their interesting properties are revealed only if their normal basis (NB) representations are considered.

1.3 Outline of Chapters

Chapter 2 deals with the general theory of GSFs. The theory of GSFs presented in this chapter has been formulated in an algebraic framework. The existing representations of GSFs are viewed through an algebraic approach. Thus Frobenius cycles in GPs representing GSFs have been regarded as members of minimal ideals in a monoid algebra. Procedures for the computation of GP coefficients and function values are discussed. Two polynomial computation techniques are discussed for function evaluation. First is based on Frobenius sum computing which is used when the GP coefficients satisfy nontrivial conjugacy relations. Second technique is based on Horner's rule which may be used for polynomial evaluation irrespective of the type of the GP under consideration.

Multi-dimensional GSFs are briefly discussed with special reference to two-dimensional (2-D) GSFs which have been used later on for decoding of linear block codes.

Chapter 3 is on the class of linearized GSFs (LGSFs), representing linear mappings, which constitutes an ideal in a monoid algebra. A LGSF is represented by a linearized Galois polynomial (LGP). An isomorphism between LGSFs and linear transformations is established. It is shown that any linear transformation from the vector space of k -tuples, $\text{GF}(2^k)$, to the vector space of n -tuples, $\text{GF}(2^n)$, k not necessarily equal to n , is

representable by a LGP whose coefficients are determined from the vectors which generate the linear transformation. Further, the conjugacy relations in these LGPs are examined. LGPs whose coefficients satisfy nontrivial conjugacy constraints have been called linearized Frobenius polynomials (LFPs) whereas those LGPs whose coefficients satisfy trivial conjugacy relations are called simply as linearized polynomials (LPs). Classes of single term LGPs which exhibit interesting algebraic structures under the operation of symbolic multiplication, are studied. It is shown that these classes possess the structure of a finite field isomorphic to $GF(2^n)$.

In Chapter 4, the monoid algebra model of GSFs is utilized for characterization, classification and synthesis of BFs. The class of BFs are represented by a monoid algebra over $GF(2)$, as a consequence of which the subclass of linear Boolean functions (LBFs) are viewed as ideals in this algebra. Generalized Reed-Muller (GRM) codes which are constructed from LBFs are also characterized in terms of ideal structures in this algebra.

Existing equivalence relations, known as the five invariance operations, for classification of BFs are examined and the effect of these operations on the corresponding GP coefficients is studied, based on which a class identification procedure for 2 and 3 variable BFs by verification of their GP coefficients is proposed. Further, a finite field model which realizes BFs based on these invariance operations is suggested. Alternately, some equivalence relations based on the monoid algebra structure of BFs are suggested and classification of two and three variable BFs is carried out based on them. It is shown that these classes contain elements which are members of ideals in a monoid algebra. A finite field model which synthesizes BFs as sum of elements from minimal ideals in a monoid algebra is suggested. This turns out to be a Frobenius sum computer, which can be efficiently realized if NB is employed. Classes of β -self dual (SD)/anti self dual (ASD) BFs are characterized using GSFs and the constraints on their GP coefficients are derived for 2, 3 and 4 variable cases.

The theory of LGSFs dealt with in Chapter 3 is applied in Chapter 5. It is shown that any LGP which represents a linear mapping and whose coefficients satisfy certain nonzero determinant property, represents a one-to-one mapping and hence a linear block code. The analogy between LGP representation and the generator matrix (basis) representation of a linear block code is brought out thus explaining the fact that there can be as many number of different LGP representations for a linear block code as the number of ways a basis can be chosen for the same.

Classes of LGSFs, not necessarily one-to-one, are studied in terms of the nature of the linear transformations generated by them. First, it is shown that by representing the transformations with respect to some NB, it is possible to group them into classes having the same weight distribution. Thus if the LGP representing one transformation in a class is known, then the LGP representations of others in that class may also be found out. Secondly, it is shown that by grouping LGSFs in a particular manner, it is possible to distinguish between one-to-one and many-to-one mappings, if the nature of at least one mapping in a class is known. Thirdly, a study of single term LGPs is attempted in detail and it is shown that any single term LGP representing a linear (n,k) transformation, $k|n$, always represents a one-to-one mapping and hence a linear block code. The nature of the codes generated by single term LGPs having the structure of a finite field (discussed in Chapter 3) are examined and a study of the distinctness of these codes, ie., finding the number of LGPs representing distinct codes in each field, is conducted. This helps partially in finding LGPs representing distinct linear block codes given some LGPs which are known to represent one-to-one mappings. It is shown that when n and k are relatively prime, all the codes generated by LGPs in such a finite field are distinct.

It is known that the roots of a LGP form a subspace. A study on the roots of LGPs representing LGSFs of linear block codes is conducted to see if they characterize individual codes. It is argued that only groups of linear block codes and not individual ones can be characterized by their roots. Further, the roots may not lie in the same field. It is also

shown that the roots cannot assume nonzero values from $GF(2^k)$.

Canonic representations of cyclic codes in terms of LGPs, both in standard basis (SB) and in normal basis (NB), are derived. It is shown that NB LGPs representing some (n,k) cyclic codes whose $k|n$, have a p -polynomial representation, ie., a LGP with coefficients from the ground field.

The role of GSFs for decoding of linear block codes is also discussed. It is shown that any standard array has a compact representation in the form of a 2-D GSF and that implementation of standard array decoders is possible using both 1-D and 2-D GSFs. The fact that a variety of options are open for the implementation of standard array decoders using both 1-D and 2-D GSFs depending on the choice of the decoded vector as a k -tuple message vector or as an n -tuple code vector, is illustrated with suitable examples. If a 1-D GSF is employed for decoding a received vector into a k -tuple message vector, then it is shown that a linear block code always has a decoder based on Frobenius sum computation and hence NB representations may be used to advantage in such situations, besides exploiting the parallelism in Frobenius polynomial representations. The chapter concludes with the GSF representation of syndrome tables connected with standard arrays and it is shown that the GSFs associated with syndrome tables are LFFs which are represented by LFPs, whose roots in $GF(2^n)$ constitute the code vectors of the corresponding linear block code.

In Chapter 6, special types of GSFs represented by LPs, called syndrome polynomials (SPs), are studied with reference to coding theory. These LPs have the property that their root space forms the code vectors of a linear block code, and the range space forms the root space of its dual LP, and vice versa. It is shown that the code vectors of any linear (n,k) block code have a unique characterization in terms of the roots of a monic LP over $GF(2^n)$ of degree 2^k , and conversely, any monic LP of degree 2^k with

nonrepetitive roots and which divides $x^{2^n} - x$, uniquely represent a linear (n, k) block code; the roots of LP constitute the code vectors. Because of this one-to-one correspondence of linear block codes and LPs, the number of LPs which divide $x^{2^n} - x$ is shown to be equal to the number of distinct linear (n, k) codes of a given pair of n and k . It is further shown that the LP whose roots form the code vectors of a linear (n, k) code can in fact be used for computing syndromes for the same (the syndromes being n -tuples instead of the usual $(n-k)$ tuples), thereby finding utility in decoding and accounting for its name.

Next, the usefulness of considering the code vectors of a linear block code as elements with respect to some NB of $GF(2^n)$ is considered. First, it is shown that this helps in identifying the normal basis syndrome polynomials (NB SPs) of linear block codes having the same weight distribution. Secondly, it is brought out that this helps in characterizing the class of t -cyclic codes (quasi cyclic codes which are closed under t cyclic shifts, $t \geq 1$) by their NB SP representations. It is shown that any linear (n, k) t -cyclic code is uniquely representable by a monic NB LP of degree 2^k with coefficients from $GF(2^t)$, where $GF(2^t)$ is a subfield of $GF(2^n)$. Thirdly, characterization of the important class of cyclic codes by their NB SP representations is given. It is shown that any linear (n, k) cyclic code has a NB SP representation in the form of a p -polynomial, i.e., a LP with coefficients from the ground field, and therefore the well known theory of p -polynomials is applicable in the characterization of cyclic codes.

A new approach to the study of the weight distributions of cyclic codes, by factorization of their NB p -polynomial representations, is formulated. It is shown that the number of irreducible polynomials in the factorization of the NB p -polynomial representing a cyclic code is equal to the number of cycles in that code, and a representative root of each irreducible polynomial in the factorization gives a cycle representative of the code. The theory of p -polynomials is applied to Bose-Chaudhuri-Hocquenghem (BCH) and Golay codes, since they are essentially cyclic

codes, and their NB p -polynomial representations and the determination of their weight distribution are illustrated with suitable examples.

Finally, the NB p -polynomial representations of self dual (n,k) cyclic codes are derived to show that they split in $GF(2^k)$, thus revealing the fact that the number of cycles in this code is equal to the number of Frobenius classes in $GF(2^k)$, with the number of members in each cycle being equal to the order of the Frobenius class. Further, their weight distribution is determined by examining the weight of a representative member of each Frobenius class of $GF(2^k)$ expressed in NB cartesian form in $GF(2^n)$.

Chapter 7 is the concluding chapter where the results of our investigation are summarized and suggestions for further research are given.

Appendix A gives the necessary mathematical background needed for understanding the theory outlined in this thesis. This appendix is divided into 3 sections. The first section briefly describes the basic algebraic structures utilized in this thesis. Section 2 is on DFT over finite fields whose extensions are used for transform domain studies in this thesis. Last section is on the theory of LPs which has been applied in the study of linear block codes.

Appendix B gives a procedure which has been developed for factorization of polynomials over finite fields using the concept of DFT over finite fields.

Appendix C gives the SB and NB finite field tables for $GF(2^n)$, for $2 \leq n \leq 5$.

CHAPTER 2

THEORY OF GALOIS SWITCHING FUNCTIONS

A combinational network with k inputs and n outputs may be represented by a set of n switching functions of k variables over $GF(2)$. Boolean algebra has been conventionally used for the analysis and synthesis of such switching functions. Use of finite fields or Galois fields for these purposes in place of Boolean algebra has been investigated by several authors. By employing finite fields, it is possible to represent a set of n functions of k variables by a single variable polynomial over an appropriate extension field of $GF(2)$. Such polynomials which we call Galois polynomials (GPs) have a well defined algebraic structure and possess remarkable properties based on Frobenius cycles, which help in the design and synthesis of switching circuits, besides aiding the construction of encoders and decoders for error control codes. The functions described by these polynomials which essentially represent mappings from $GF(2^k)$ to $GF(2^n)$ are called one-dimensional Galois switching functions (1-D GSFs) or simply Galois switching functions (GSFs). In general, finite fields can be of characteristic p ; p a prime number. However, we consider only the case $p = 2$.

The concept of 1-D GSFs described by single variable polynomials may be extended to multi-dimensional GSFs, especially two dimensional (2-D) GSFs described by two-variable GPs, and their properties may be studied. The 2-D GSFs are particularly useful for representation and implementation of standard arrays used for decoding of linear block codes. Besides, they may be effectively employed for representation and processing images (pictorial data) represented in the form of two variable GPs.

This chapter discusses the general theory of GSFs. Algebraic models for GSFs are suggested. Frobenius properties of GPs representing GSFs are discussed. Computational techniques for coefficients as well as function values of a GP are presented. Finally, multi-dimensional GSFs, with special reference to 2-D GSFs, are studied.

Hereafter, we use the term 'GSF' for 1-D Galois switching functions unless otherwise stated.

2.1 Notion of a Galois Switching Function (GSF)

To begin with we introduce the notion of GSFs. Towards this end consider, as an example, a system of $n = 2$ Boolean functions (BFs) of $k = 3$ variables given by the conventional truth table of Table 2.1. In this table, the output of the system is listed with the input (binary 3-tuples) arranged in ascending order 0, 1, 2, ..., 7 (in general, 0, 1, 2, ..., $2^k - 1$). This order is called the *natural order*.

Table 2.1: Truth Table Representation of a System of 2 Boolean Functions of 3 Variables

Input			Output	
x_2	x_1	x_0	y_1	y_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

Any truth table correspondence $(x_{k-1}, \dots, x_1, x_0)$ to $(y_{n-1}, \dots, y_1, y_0)$ can

also be expressed in terms of appropriate finite field elements. In this connection, note that any non-zero element of a finite field, say $GF(2^k)$, can be represented by a power α^j (polar form) of a primitive element α (of $GF(2^k)$), and at the same time this can be represented by a polynomial of α

$$\zeta_{k-1} \alpha^{k-1} + \zeta_{k-2} \alpha^{k-2} + \dots + \zeta_1 \alpha + \zeta_0 \quad ; \quad \zeta_i \in GF(2). \quad (2.1.1)$$

if the minimal polynomial over $GF(2)$ of α is given. Thus we can take the set of coefficients of (2.1.1)

$$\zeta_{k-1}, \zeta_{k-2}, \dots, \zeta_0$$

as cartesian representation of finite field elements. Accordingly the contents of any conventional truth table can be interpreted and expressed in terms of appropriate field elements.

With reference to the example under consideration, let α be a primitive element of $GF(2^3)$ and β be a primitive element of $GF(2^2)$. If we take the minimal polynomial of α to be $x^3 + x^2 + 1$ and the minimal polynomial of β to be $x^2 + x + 1$, then the mapping given in Table 2.1, interpreted as a mapping from $GF(2^3)$ to $GF(2^2)$, becomes as shown in Table 2.2.

Finally, the mapping of Table 2.2 can be rearranged such that the input index set corresponds to increasing powers of α (this order being called the *field order*), resulting in the mapping shown in Table 2.3 which defines a typical Galois switching function (GSF).

In general, GSFs can be defined as mappings from one finite field $GF(p^k)$ to another finite field $GF(p^n)$, where p is a prime number and integer k is not necessarily equal to integer n , and the input index set (domain of the mapping) is ordered to correspond to increasing powers of a primitive element of $GF(2^k)$. In this thesis we restrict ourselves to the practically important case $p = 2$. The definition of a GSF given here will be refined in the next section where we consider them in an algebraic framework.

Table 2.2: Truth Table of the System of Boolean Functions of Table 2.1 Considered as a Mapping from $GF(2^3)$ to $GF(2^2)$ with the Input Variables in Natural Order

Input				Output			
polar form x		cartesian form x ₂ x ₁ x ₀		cartesian form y ₁ y ₀		polar form y	
0		0 0 0		0 0		0	
1		0 0 1		0 1		1	
α		0 1 0		1 1		β ²	
α ⁵		0 1 1		1 0		β	
α ²		1 0 0		1 0		β	
α ³		1 0 1		0 1		1	
α ⁶		1 1 0		1 0		β	
α ⁴		1 1 1		0 1		1	

Table 2.3: Truth Table of the Mapping Given in Table 2.2 with the Input Variables in Field Order

Input				Output			
polar form x		cartesian form $x_2 \ x_1 \ x_0$		cartesian form $y_1 \ y_0$		polar form y	
0		0	0 0	0 0		0	
1		0	0 1	0 1		1	
α		0	1 0	1 1		β^2	
α^2		1	0 0	1 0		β	
α^3		1	0 1	0 1		1	
α^4		1	1 1	0 1		1	
α^5		0	1 1	1 0		β	
α^6		1	1 0	1 0		β	

2.2 Algebraic Models for Galois Switching Functions (GSFs)

We continue our discussion on GSFs by describing the algebraic models of signals given in [12]. The equivalence between these models and the one given in [10] will be brought out in Section 2.3 to show that the existing representations of GSFs can be studied in an algebraic framework.

2.2.1 Monoid Algebra Model of GSFs

In the previous section, GSFs have been defined essentially as mappings from one finite field $GF(2^k)$ to another finite field $GF(2^n)$. Let us closely examine the structure of the index set $GF(2^k)$ over which the GSFs are defined. The nonzero elements of $GF(2^k)$ constitute a multiplicative cyclic group of order 2^k-1 . However, the multiplicative inverse of the element 0 of $GF(2^k)$ is not defined. Under the multiplication operation, the elements of $GF(2^k)$ accordingly have the structure of a *cyclic monoid* (for definition and properties of monoids see Appendix A). We denote this monoid by $M(2^k)$.

For notational uniformity, we denote the element 0 of $GF(2^k)$ and $M(2^k)$ by $\alpha^{-\infty}$, where α is a primitive element of $GF(2^k)$. Thus the elements of $GF(2^k)$ as well as $M(2^k)$ are $\alpha^{-\infty}, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^\xi$, where $\xi = 2^k-2$.

Consider signals defined as functions on a finite index set with the structure of the finite multiplicative cyclic monoid $M(2^k)$. If we denote the value of such a function \underline{f} at the index α^i by f_{α^i} , the function takes the form of the finite sequence

$$\underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^0}, f_{\alpha^1}, \dots, f_{\alpha^\xi}); \xi = 2^k-2. \quad (2.2.1)$$

With the above discussion in view GSFs, such as defined in Section 2.1, can now be viewed as *signals or functions on a finite index set having the structure of a multiplicative cyclic monoid, say $M(2^k)$, and taking their values from a finite field, say $GF(2^n)$.*

It may be noted that the set of all GSFs defined over $M(2^k)$ and assuming values from a finite field $F = GF(2^n)$ has the structure of a *vector space* over F of dimension 2^k .

This vector space is denoted by F^{2^k} .

Consider the set of all signals with domain $M(2^k)$ and range $GF(2^n)$.

Let $\underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^0}, f_{\alpha^1}, \dots, f_{\alpha^\zeta})$; $\xi = 2^k - 2$

and $\underline{s} = (s_{\alpha^{-\infty}}, s_{\alpha^0}, s_{\alpha^1}, \dots, s_{\alpha^\zeta})$; $\xi = 2^k - 2$

be two elements of this set. We define two operations on them as follows:

(i) pointwise addition (+) of \underline{f} and \underline{s} , denoted by $\underline{f} + \underline{s}$, produces another element from the set given by

$$\underline{f} + \underline{s} = (f_{\alpha^{-\infty}} + s_{\alpha^{-\infty}}, f_{\alpha^0} + s_{\alpha^0}, f_{\alpha^1} + s_{\alpha^1}, \dots, f_{\alpha^\zeta} + s_{\alpha^\zeta}) \quad (2.2.2)$$

and

(ii) convolution (*) of \underline{f} and \underline{s} , denoted by $\underline{f} * \underline{s} (= \underline{s} * \underline{f})$, produces another element \underline{g} from the set whose components $g_{\alpha^{-\infty}}, g_{\alpha^0}, g_{\alpha^1}, \dots, g_{\alpha^\zeta}$ are given by

$$g_{\alpha^{-\infty}} = f_{\alpha^{-\infty}} s_{\alpha^{-\infty}} \quad (2.2.3a)$$

and

$$g_{\alpha^i} = \sum_{j=0}^{\zeta} s_{\alpha^{i \oplus j}} f_{\alpha^j} ; \quad \xi = 2^k - 2, i = 0, 1, \dots, \xi, \quad (2.2.3b)$$

The set of all signals over the index set $M(2^k)$ with the two binary operations of pointwise addition and convolution, as given by (2.2.2) and (2.2.3) respectively, can be seen to have the structure of a commutative algebra which will be called a *cyclic monoid algebra* of dimension 2^k .

The GSFs may now be interpreted to be members of a cyclic monoid algebra with domain $M(2^k)$ and range $GF(2^n)$.

The convolution matrix S , with respect to the standard basis, has the following block diagonal structure

$$S = \begin{bmatrix} s_{\alpha^{-\infty}} & 0 & 0 & \cdot & \cdot & 0 \\ 0 & s_{\alpha^0} & s_{\alpha^\zeta} & \cdot & \cdot & s_{\alpha^1} \\ 0 & s_{\alpha^1} & s_{\alpha^0} & \cdot & \cdot & s_{\alpha^2} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & s_{\alpha^\zeta} & s_{\alpha^{\zeta-1}} & \cdot & \cdot & s_{\alpha^0} \end{bmatrix} \quad (2.2.4)$$

where $\zeta = 2^k - 2$, and the matrix formed by deleting the first row and first column of S , say S_c , is a cyclic matrix of order $2^k - 1$ with

$$(s_{\alpha^0}, s_{\alpha^1}, \dots, s_{\alpha^\zeta}); \quad \zeta = 2^k - 2$$

as its generating vector.

The block diagonal structure of the convolution matrix S may be compactly expressed as

$$S = \left[\begin{array}{c|c} s_{\alpha^{-\infty}} & \\ \hline & S_c \end{array} \right], \quad (2.2.5)$$

where the blank spaces denote zeroes.

One of the important consequences of the above structure of the convolution matrix is described below:

Define the following set of $(2^k - 1)$ permutations on f :

$$(P_{\alpha^j} f)_{\alpha^{-\infty}} = f_{\alpha^{-\infty}} \quad (2.2.6a)$$

$$i, j = 0, 1, \dots, 2^k - 2$$

$$(P_{\alpha^j} f)_{\alpha^i} = f_{\alpha^{i \ominus j}}, \quad (2.2.6b)$$

where \ominus stands for subtraction modulo $2^k - 1$. The relations in (2.2.6) may be expanded as

$$P_{\alpha^0} f = (f_{\alpha^{-\infty}}, f_{\alpha^0}, f_{\alpha^1}, \dots, f_{\alpha^\zeta})$$

$$P_{\alpha^1} f = (f_{\alpha^{-\infty}}, f_{\alpha^\zeta}, f_{\alpha^0}, \dots, f_{\alpha^{\zeta-1}})$$

$$P_{\alpha^j} \underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^1}, f_{\alpha^2}, \dots, f_{\alpha^0}), \text{ where } j = 2^k - 2.$$

Thus in the case of GSFs over $M(2^2)$ we have

$$P_{\alpha^0} \underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^0}, f_{\alpha^1}, f_{\alpha^2})$$

$$P_{\alpha^1} \underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^2}, f_{\alpha^0}, f_{\alpha^1})$$

$$P_{\alpha^2} \underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^1}, f_{\alpha^2}, f_{\alpha^0}).$$

It is noted from the above relations that these permutations have the effect of keeping the value of \underline{f} at the index $\alpha^{-\infty}$ fixed, and cyclically permuting the remaining values towards the right by j positions.

It is easy to verify that the set $\{P_{\alpha^j}\}$ of all the permutations as defined above has the structure of a cyclic group of order $(2^k - 1)$.

The input-output pairs for the class of linear transformations from F^{2^k} to F^{2^k} , defined by the convolution operation given by (2.2.3), preserve the effect of the permutations defined by (2.2.6) (A linear transformation is said to preserve the effect of permutations on the input signal, if the output of the same is permuted in the same manner as the input [31]). Thus, if S is any member of such a class of linear transformations with input and output signals \underline{f} and \underline{g} respectively, i.e.,

$$\underline{g} = S \underline{f} \tag{2.2.7}$$

then we have

$$S P_{\alpha^j} \underline{f} = P_{\alpha^j} S \underline{f} ; j = 0, 1, 2, \dots, 2^k - 2. \tag{2.2.8}$$

2.2.2 Transform Domain Description of GSFs

We now proceed to obtain an appropriate transform for describing the GSFs belonging to a monoid algebra. Towards this end, we take note of the fact that any cyclic

matrix can be diagonalized by a discrete Fourier transform (DFT) matrix of the same order. So if we take a matrix H of the form

$$H = \left[\begin{array}{c|c} 1 & \\ \hline & H_d \end{array} \right], \quad (2.2.9)$$

where H_d is a DFT matrix of order $2^k - 1$ over an appropriate extension field, then it may be easily seen that the inverse of the matrix H , say, H^{-1} , diagonalizes the block diagonal convolution matrix S defined in (2.2.5). ie., $H S H^{-1} = \underline{\Lambda}$ where $\underline{\Lambda}$ is a diagonal matrix, whose first diagonal entry is $s_{-\infty} = \Lambda_{-\infty}$, (say), and the remaining diagonal entries can be shown to be the DFT coefficients of the generating vector of the cyclic matrix S_c given by its first column.

It should be noted that H^{-1} also diagonalizes all the permutation matrices of the permutation operators $P_{\alpha j}$, $j = 0, 1, 2, \dots, 2^k - 2$.

For GSFs belonging to the monoid algebra, we can now define a transform pair, $\underline{f} \longrightarrow \underline{F}$, as follows:

$$\begin{aligned} \underline{F} &= H \underline{f} \\ \text{and} \quad \underline{f} &= H^{-1} \underline{F}, \end{aligned} \quad (2.2.10)$$

where $\underline{F} = (F_{\alpha^{-\infty}}, F_{\alpha^0}, F_{\alpha^1}, F_{\alpha^2}, \dots, F_{\alpha^{\xi}})$; $\xi = 2^k - 2$.

Consider the convolution of two GSFs \underline{f} and \underline{g} , as defined by (2.2.3). Let $\underline{g} = \underline{f} * \underline{g}$. This relation in matrix form is $\underline{g} = S \underline{f}$, where S has the structure of the matrix of (2.2.5). Multiplying by H on both sides, we get

$$H \underline{g} = H S \underline{f}$$

which may be written as

$$\underline{G} = H S H^{-1} H \underline{f}$$

or

$$\underline{G} = \underline{\Lambda} \underline{F},$$

where $f \longmapsto \underline{F}$, $g \longmapsto \underline{G}$ and $s \longmapsto \underline{\Lambda} = H S H^{-1}$ define three transform pairs. Since, $\underline{\Lambda}$ is a diagonal matrix with $\Lambda_{\alpha i}$, $i = -\infty, 0, 1, \dots, 2^k-2$, as the diagonal elements, it follows that

$$G_{\alpha i} = \Lambda_{\alpha i} F_{\alpha i}; \quad i = -\infty, 0, 1, \dots, 2^k-2.$$

Thus the operation of convolution in the function domain gets translated to the operation of pointwise multiplication in the transform domain. Formally, we have

Convolution Theorem : Let $f \longmapsto \underline{F}$ and $s \longmapsto \underline{\Lambda}$ be two transform pairs, then

$$(f * s) \longmapsto (\underline{F} \cdot \underline{\Lambda})$$

and

$$(\underline{F} * \underline{\Lambda}) \longmapsto (f \cdot s)$$

where '*' stands for the convolution operation as defined in (2.2.3) and '.' stands for pointwise product.

If we represent the functions f and s , in terms of their transform coefficients \underline{F} and $\underline{\Lambda}$ respectively, then the set of all transformed signals, with the following two binary operations defined on them in the transform domain:

(i) pointwise addition $\underline{F} + \underline{\Lambda}$:

$$F_{\alpha i} + \Lambda_{\alpha i}, \quad i = -\infty, 0, 1, 2, \dots, 2^k-2. \quad (2.2.11)$$

(ii) pointwise multiplication $\underline{F} \cdot \underline{\Lambda}$:

$$F_{\alpha i} \cdot \Lambda_{\alpha i}, \quad i = -\infty, 0, 1, 2, \dots, 2^k-2. \quad (2.2.12)$$

constitutes a commutative algebra of dimension 2^k . This algebra is, in fact, isomorphic to the cyclic monoid algebra introduced earlier; the transform defined in (2.2.10) acting as the isomorphism between these two algebras.

Transform domain description of cyclic monoid algebra leads to an alternative characterization of GSFs in terms of finite field polynomials, to be called Galois polynomials (GPs).

2.3 Galois Polynomial Representation of GSFs

In this section we show that GSFs can be represented by single variable polynomials over appropriate extensions of finite fields. Such a polynomial representation of GSFs will be seen to be the one given by Takahashi [10]. A polynomial representing a GSF is called a Galois polynomial (GP). The set of all GPs constitutes a residue class polynomial algebra with suitable polynomial operations defined on them.

A polynomial expression, say $f(x)$, for a GSF \underline{f} may be obtained in which the value of the polynomial $f(x)$ at $x = \alpha^j$ is set equal to the value of the GSF at the index α^j . Key results on GSFs that follow from such an approach are given below.

(a) Consider the convolution matrix of the form (2.2.5) for signals with domain and range as $M(2^k)$ and $GF(2^n)$ respectively. Eigenvectors of the convolution matrix, as given by the columns of the matrix H^{-1} , are specific examples of GSFs. They can be described by the following polynomials

$$h_{\alpha^{-\infty}}(x) = (x^{2^k-1})^\dagger$$

$$h_{\alpha^i}(x) = x^{2^k-1-i} ; i = 0, 1, \dots, 2^k-2,$$

where $(x^{2^k-1})^\dagger$ denotes the complement of (x^{2^k-1}) . The value of $h_{\alpha^i}(x)$ for $x = \alpha^j$ equals the value of the $(\alpha^j)^{th}$ eigenvector at the index α^j .

(b) The polynomial representation of eigenvectors leads to a polynomial representation of any GSF in a natural manner. Using the relation $\underline{f} = H^{-1}\underline{F}$, we obtain a polynomial expression $f(x)$ given below

$$f(x) = F_{\alpha^{-\infty}}(x^{2^k-1})^\dagger + \sum_{i=0}^{2^k-2} F_{\alpha^i} x^{2^k-1-i}$$

The value of $f(x)$ for $x = \alpha^i$ is equal to the value of the GSF at the index α^i . Note that $f(\alpha^{-\infty}) = f(0) = F_{\alpha^{-\infty}}$.

Since $(x^{2^k-1})^\dagger = \begin{cases} 1 & \text{for } x = \alpha^{-\infty} \\ 0 & \text{for } x = \alpha^j, j = 0, 1, \dots, 2^k-2 \end{cases}$

we may write $(x^{2^k-1})^\dagger = 1 + (x^{2^k-1})$. Thus $f(x)$ may be written as

$$f(x) = F_{\alpha^{-\infty}} + (F_{\alpha^{-\infty}} + F_{\alpha^0}) x^{2^k-1} + \sum_{i=1}^{2^k-2} F_{\alpha^i} x^{2^k-1-i}.$$

or
$$f(x) = a_{\alpha^{-\infty}} + a_{\alpha^0} x^{2^k-1} + \dots + \sum_{i=1}^{2^k-2} a_{\alpha^i} x^{2^k-1-i}. \quad (2.3.1)$$

where $a_{\alpha^{-\infty}} = F_{\alpha^{-\infty}}$, $a_{\alpha^0} = (F_{\alpha^{-\infty}} + F_{\alpha^0})$, and $a_{\alpha^i} = F_{\alpha^i}$, $i = 1, 2, \dots, 2^k-2$.

Expression (2.3.1) provides the desired polynomial representation of a GSF. A polynomial of the form (2.3.1) representing a GSF f will be called a Galois polynomial (GP). It may be seen that the GP representation of a GSF is of the same form as that given in [10].

(c) The coefficient vector

$$\underline{a} = (a_{\alpha^{-\infty}}, a_{\alpha^0}, a_{\alpha}, \dots, a_{\alpha^\xi}); \xi = 2^k-2$$

and the function value vector

$$\underline{f} = (f_{\alpha^{-\infty}}, f_{\alpha^0}, f_{\alpha}, \dots, f_{\alpha^\xi}); \xi = 2^k-2,$$

can be seen to be related through a nonsingular $2^k \times 2^k$ matrix given below.

$$\begin{bmatrix} f(0) \\ f(1) \\ f(\alpha) \\ f(\alpha^2) \\ \vdots \\ f(\alpha^\xi) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdot & \cdot & 0 & 0 \\ 1 & 1 & 1 & \cdot & \cdot & 1 & 1 \\ 1 & 1 & (\alpha^\xi) & \cdot & \cdot & \alpha^2 & \alpha \\ 1 & 1 & (\alpha^\xi)^2 & \cdot & \cdot & \alpha^4 & \alpha^2 \\ 1 & 1 & (\alpha^\xi)^3 & \cdot & \cdot & \alpha^6 & \alpha^3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \alpha & \cdot & \cdot & \alpha^\xi & \cdot \end{bmatrix} \begin{bmatrix} a_{\alpha^{-\infty}} \\ a_{\alpha^0} \\ a_{\alpha} \\ a_{\alpha^2} \\ \cdot \\ \cdot \\ a_{\alpha^\xi} \end{bmatrix}$$

from which we get

$$\begin{bmatrix} a_{\alpha^{-\infty}} \\ a_{\alpha^0} \\ a_{\alpha} \\ . \\ . \\ . \\ a_{\alpha^{\xi}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & 0 \\ 1 & 1 & 1 & 1 & . & . & 1 \\ 0 & 1 & \alpha & \alpha^2 & . & . & (\alpha)^{\xi} \\ 0 & 1 & \alpha^2 & \alpha^4 & . & . & (\alpha^2)^{\xi} \\ 0 & 1 & \alpha^3 & \alpha^6 & . & . & (\alpha^3)^{\xi} \\ . & . & . & . & . & . & . \\ 0 & 1 & \alpha^{\xi} & . & . & . & \alpha \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ f(\alpha) \\ f(\alpha^2) \\ . \\ . \\ f(\alpha^{\xi}) \end{bmatrix}$$

or

$$\underline{a} = \mathcal{G} \underline{f} \quad (2.3.2)$$

where \underline{a} is the coefficient vector on the left hand side (LHS), \underline{f} is the function value vector on the right hand side (RHS), and the matrix \mathcal{G} is the $2^k \times 2^k$ matrix on the RHS. The function value vector can be obtained from the coefficient vector by

$$\underline{f} = \mathcal{G}^{-1} \underline{a} \quad (2.3.3)$$

Equations (2.3.2) and (2.3.3) define a transform pair for GSFs which we will call *Galois transform (GT) pair*; the matrix \mathcal{G} will be referred to as *Galois transform matrix*.

(d) The GT relationship (2.3.2) in a slightly modified form can be expressed as follows:

$$a_{\alpha^{-\infty}} = f_{\alpha^{-\infty}}$$

and

$$\begin{bmatrix} a_{\alpha^0} & a_{\alpha^{-\infty}} \\ a_{\alpha} \\ a_{\alpha^2} \\ \vdots \\ a_{\alpha^{\xi}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \alpha & \alpha^2 & \dots & \dots & \alpha^{\xi} \\ 1 & \alpha^2 & \alpha^4 & \dots & \dots & (\alpha^2)^{\xi} \\ 1 & \alpha^3 & \alpha^6 & \dots & \dots & (\alpha^3)^{\xi} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha^{\xi} & \dots & \dots & \dots & \alpha \end{bmatrix} \begin{bmatrix} f(1) \\ f(\alpha) \\ f(\alpha^2) \\ f(\alpha^3) \\ \vdots \\ f(\alpha^{\xi}) \end{bmatrix} \quad (2.3.4)$$

where $\xi = 2^k - 2$ and the $2^{k-1} \times 2^{k-1}$ matrix on the RHS can be recognized as a DFT matrix. Thus the coefficients are related to the function values by a DFT relation.

2.3.1 Residue Class Polynomial Algebra Model of GSFs

Let $f(x) = a_{\alpha^{-\infty}} + a_{\alpha^0} x^{2^{k-1}} + a_{\alpha} x^{2^{k-2}} + \dots + a_{\alpha^{\xi}} x$; $\xi = 2^k - 2$

and $s(x) = \Lambda_{\alpha^{-\infty}} + \Lambda_{\alpha^0} x^{2^{k-1}} + \Lambda_{\alpha} x^{2^{k-2}} + \dots + \Lambda_{\alpha^{\xi}} x$; $\xi = 2^k - 2$

be two GPs representing GSFs \underline{f} and \underline{g} respectively. We define two binary operations on them as follows:

(i) Polynomial addition :

$$f(x) + s(x) = (a_{\alpha^{-\infty}} + \Lambda_{\alpha^{-\infty}}) + (a_{\alpha^0} + \Lambda_{\alpha^0}) x^{2^{k-1}} + \dots + (a_{\alpha^{\xi}} + \Lambda_{\alpha^{\xi}}) x \quad (2.3.5)$$

(ii) Polynomial multiplication $f(x)s(x)$ modulo $(x^{2^k} + x)$.

It may be verified that the set of all GPs of the kind given above with the two binary operations of polynomial addition and polynomial multiplication modulo $(x^{2^k} + x)$ constitutes a commutative algebra of dimension 2^k over the field $F = GF(2^n)$. This algebra is called *residue class polynomial algebra modulo $(x^{2^k} + x)$* .

Let $m(x) = f(x)s(x)$ modulo $(x^{2^k} + x)$. Then $m(x)$ can also be expressed in the form:

$$m(x) = M_{\alpha^{-\infty}} + M_{\alpha^0} x^{2^k-1} + M_{\alpha} x^{2^k-2} + \dots + M_{\alpha^\zeta} x ; \quad \zeta = 2^k-2.$$

It can be verified that the coefficients of GPs $f(x)$, $s(x)$ and $m(x)$ are related by the convolutional relationship $\underline{M} = \underline{F} * \underline{\Lambda}$ of the kind given by

$$M_{\alpha^{-\infty}} = a_{\alpha^{-\infty}} \Lambda_{\alpha^{-\infty}} \quad (2.3.6a)$$

and

$$\begin{bmatrix} M_{\alpha^0} - M_{\alpha^{-\infty}} \\ M_{\alpha} \\ M_{\alpha^2} \\ \vdots \\ M_{\alpha^\zeta} \end{bmatrix} = \begin{bmatrix} a_{\alpha^0} - a_{\alpha^{-\infty}} \\ a_{\alpha} \\ a_{\alpha^2} \\ \vdots \\ a_{\alpha^\zeta} \end{bmatrix} \otimes \begin{bmatrix} \Lambda_{\alpha^0} - \Lambda_{\alpha^{-\infty}} \\ \Lambda_{\alpha} \\ \Lambda_{\alpha^2} \\ \vdots \\ \Lambda_{\alpha^\zeta} \end{bmatrix} \quad (2.3.6b)$$

where $\zeta = 2^k-2$ and \otimes denotes cyclic convolution.

Thus for $k = 2$, we have

$$\begin{bmatrix} M_{\alpha^0} - M_{\alpha^{-\infty}} \\ M_{\alpha} \\ M_{\alpha^2} \end{bmatrix} = \begin{bmatrix} a_{\alpha^0} - a_{\alpha^{-\infty}} & a_{\alpha^2} & a_{\alpha} \\ a_{\alpha} & a_{\alpha^0} - a_{\alpha^{-\infty}} & a_{\alpha^2} \\ a_{\alpha^2} & a_{\alpha} & a_{\alpha^0} - a_{\alpha^{-\infty}} \end{bmatrix} \begin{bmatrix} \Lambda_{\alpha^0} - \Lambda_{\alpha^{-\infty}} \\ \Lambda_{\alpha} \\ \Lambda_{\alpha^2} \end{bmatrix}$$

3.2 Modified Cyclic Monoid Algebra Model for GSFs

(a) We take note of the fact that the Galois transform matrix \mathcal{G} given by

$$\mathcal{G} = \left[\begin{array}{c|c} 1 & \\ \hline 1 & H_d \end{array} \right], \quad (2.3.7)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & . & . & 0 \\ 1 & 1 & 1 & 1 & . & . & 1 \\ 0 & 1 & \alpha & \alpha^2 & . & . & (\alpha)^\xi \\ 0 & 1 & \alpha^2 & \alpha^4 & . & . & (\alpha^2)^\xi \\ 0 & 1 & \alpha^3 & \alpha^6 & . & . & (\alpha^3)^\xi \\ . & . & . & . & . & . & . \\ 0 & 1 & \alpha^\xi & . & . & . & \alpha \end{bmatrix}$$

can be obtained from the block diagonalized matrix H of (2.2.9), by putting a 1 in the first location of the second row. This modifies the coefficient F_{α^0} which is now equal to the sum of all the function values including $f(\alpha^{-\omega})$ (F_{α^0} can be said to serve as a parity check) unlike the former case where $f(\alpha^{-\omega})$ is delinked from the remaining function values. We will hereafter use the Galois transform matrix \mathcal{F} for studying GSFs in the transform domain. The inverse of the Galois transform matrix \mathcal{F}^{-1} is given by

$$\mathcal{F}^{-1} = \left[\begin{array}{c|c} \begin{matrix} 1 \\ 1 \\ \vdots \\ 1 \end{matrix} & \begin{matrix} \\ \\ H_d^{-1} \\ \end{matrix} \end{array} \right], \quad (2.3.8)$$

$$= \begin{bmatrix} 1 & 0 & 0 & . & . & 0 & 0 \\ 1 & 1 & 1 & . & . & 1 & 1 \\ 1 & 1 & (\alpha^\xi) & . & . & \alpha^2 & \alpha \\ 1 & 1 & (\alpha^\xi)^2 & . & . & \alpha^4 & \alpha^2 \\ 1 & 1 & (\alpha^\xi)^3 & . & . & \alpha^6 & \alpha^3 \\ . & . & . & . & . & . & . \\ 1 & 1 & \alpha & . & . & . & \alpha^\xi \end{bmatrix}$$

(b) With the above modification in the nature of the transform matrix, the convolutional relationship in the function domain takes the following form:

$$g = \underline{f} * \underline{s} = \underline{s} * \underline{f}$$

$$\text{with } g_{\alpha^{-\infty}} = f_{\alpha^{-\infty}} \cdot s_{\alpha^{-\infty}}, \quad (2.3.9a)$$

$$\text{and } g_{\alpha^i} = b f_{\alpha^{-\infty}} + c s_{\alpha^{-\infty}} + \sum_{j=0}^{2^k-2} s_{\alpha^{i \oplus j}} f_{\alpha^j}; i = 0, 1, \dots, 2^k-2 \quad (2.3.9b)$$

$$\text{where } b = \sum_i s_{\alpha^i}, i = 0, 1, \dots, 2^k-2, \quad (2.3.10)$$

$$\text{and } c = \sum_i f_{\alpha^i}, i = 0, 1, \dots, 2^k-2, \quad (2.3.11)$$

(c) The corresponding convolution matrix S which may be diagonalized by \mathcal{F}^1 and which satisfies the permutation preserving property defined in Section 2.2.1, for example, for $k = 2$, is shown below:

$$S = \begin{bmatrix} s_{\alpha^{-\infty}} & 0 & 0 & 0 \\ b & s_{\alpha^0+s_{\alpha^{-\infty}}} & s_{\alpha^2+s_{\alpha^{-\infty}}} & s_{\alpha+s_{\alpha^{-\infty}}} \\ b & s_{\alpha+s_{\alpha^{-\infty}}} & s_{\alpha^0+s_{\alpha^{-\infty}}} & s_{\alpha^2+s_{\alpha^{-\infty}}} \\ b & s_{\alpha^2+s_{\alpha^{-\infty}}} & s_{\alpha+s_{\alpha^{-\infty}}} & s_{\alpha^0+s_{\alpha^{-\infty}}} \end{bmatrix}$$

$$\text{where } b = s_{\alpha^0} + s_{\alpha^1} + s_{\alpha^2}.$$

(d) The set of all signals over the index set $M(2^k)$ with the two binary operations of pointwise addition and convolution as given by (2.3.9), can be seen to have the structure of a commutative algebra which may be called a *modified cyclic monoid algebra*, of dimension 2^k . This algebra is isomorphic to the residue class polynomial algebra modulo $(x^{2^k} + x)$

given in Section 2.3.1.

In what follows the GSFs will be interpreted to be members of the cyclic monoid algebra (as modified above) with domain $M(2^k)$ and range $GF(2^n)$.

(e) Consider two GSFs f and g and their convolution $f * g$ as defined by (2.3.9). The coefficients of the polynomial $g(x)$ representing g are related to the coefficients in the polynomial representation of f and g as given below.

$$G_{\alpha^i} = F_{\alpha^i} S_{\alpha^i} ; \quad i = -\infty, 0, 1, \dots, 2^k - 2. \quad (2.3.12)$$

2.3.3 Takahashi's Representation of GSFs

We have seen in the beginning of Section 2.3 that any GSF can be represented by a GP of the form (2.3.1). We have also shown that the coefficients of the GP can be evaluated by (2.3.2). Here we give Takahashi's [10] approach for obtaining coefficients of GPs representing GSFs. This approach is equivalent to the approach based on cyclic monoid algebra. The relations given in [10] may accordingly be viewed in the monoid algebraic framework.

Consider a GSF f represented by the following GP given earlier in (2.3.1) as

$$f(x) = a_{\alpha^{-\infty}} + a_{\alpha^0} x^{2^k-1} + a_{\alpha^1} x^{2^k-2} + \dots + a_{\alpha^i} x^{2^k-1-i} + \dots + a_{\alpha^\xi} x,$$

where $\xi = 2^k - 2$, $x \in GF(2^k)$, $f(x) \in GF(2^n)$ (k not necessarily equal to n), coefficients $\in GF(2^L)$, and L is the L.C.M of n and k . The coefficients of the GP given above will be shown to be given by

$$\begin{aligned} a_{\alpha^{-\infty}} &= f(0) = f(\alpha^{-\infty}) \\ \text{and } a_{\alpha^i} &= \sum_{x \in GF(2^k)} x^i f(x), \quad i = 0, 1, 2, \dots, 2^k - 2. \end{aligned} \quad (2.3.13)$$

The coefficient $a_{\alpha^{-\infty}}$ is obtained on substituting $x = \alpha^{-\infty} = 0$ in (2.3.1).

For obtaining the remaining coefficients a_{α^i} , $i = 0, 1, \dots, 2^k - 2$, we multiply (2.3.1) throughout by x^i and sum over all $x \in GF(2^k)$, to get

$$\sum x^i f(x) = a_{\alpha^\infty} \sum x^i + a_{\alpha^0} \sum x^{2^k-1+i} + a_{\alpha} \sum x^{2^k-2+i} + \dots + a_{\alpha^i} \sum x^{2^k-1} + \dots + a_{\alpha^\zeta} \sum x^{1+i}.$$

Since the sum of all the elements of $GF(2^k)$ is zero, all the terms on the RHS of the above equation except the term $a_{\alpha^i} \sum x^{2^k-1}$, vanish. Now $a_{\alpha^i} \sum x^{2^k-1} = a_{\alpha^i}$, because

$$\sum_{x \in GF(2^k)} x^{2^k-1} = 1. \text{ Thus, } a_{\alpha^i} = \sum_{x \in GF(2^k)} x^i f(x), i = 0, 1, 2, \dots, 2^k-2.$$

The following well known result in finite field theory may be used at this point to prove that the coefficients belong to $GF(2^L)$, L being the L.C.M of n and k :

$GF(2^L)$ has one and only one subfield $GF(2^m)$ iff $m|L$. Further, if γ is a primitive element of $GF(2^L)$, then β is a primitive element of $GF(2^m)$ where $\beta = \gamma^\nu$, $\nu = 2^L-1/2^m-1$.

Therefore if $n \neq k$, then we have to work in a larger field, say $GF(2^L)$, which contains both $GF(2^k)$ and $GF(2^n)$ as its subfields. If we choose L as the L.C.M of n and k , then $GF(2^L)$ contains unique subfields $GF(2^k)$ and $GF(2^n)$, since k and n both divide L . Thus the coefficients can be chosen from $GF(2^L)$. Q.E.D.

Note: In all our future discussions on GSFs, we drop the term ' α ' from their index sets for notational convenience. Thus hereafter, the function f which was denoted earlier as $(f_{\alpha^\infty} f_{\alpha^0} f_{\alpha^1} \dots, f_{\alpha^\zeta})$ will now be denoted simply as $(f_{-\infty} f_0 f_1 \dots, f_\xi)$. A similar change in notation will be adopted for the GP coefficients representing GSFs also, which will now be denoted as $(a_{-\infty}, a_0, a_1, \dots, a_\xi)$, where $\xi = 2^k-2$.

In the next section, we discuss Frobenius cycles in GPs representing GSFs:

2.4 Frobenius Cycles in GPs

When a finite field $GF(2^L)$ has a subfield $GF(2^n)$, then the transformation

$$\Theta \rightarrow \Theta^Q \tag{2.4.1}$$

is defined as a *Frobenius transformation* where $\Theta \in GF(2^L)$ and $Q = 2^n$.

Now, if Θ also belongs to $GF(2^n)$, then $\Theta^Q = \Theta$, ie., Θ remains invariant by

Frobenius transformation.

Thus Frobenius cycles exist only if $GF(2^n)$ is a proper subfield of $GF(2^L)$.

If $\Theta \in GF(2^L)$, but does not belong to its subfield $GF(2^n)$, then

$\{\Theta, \Theta^Q, \Theta^{Q^2}, \dots, \Theta^{Q^{i-1}}\}$ is defined as a *Frobenius cycle* if $\Theta^{Q^i} = \Theta$ and $\Theta^{Q^j} \neq \Theta$ for $j < i$.

GSFs have remarkable properties connected with Frobenius cycles. From the above discussion, we observe that Frobenius cycles exist only when the field to which the function values of the GSF belong, say $GF(2^n)$, is a proper subfield of the field to which the coefficients of the GP representing the GSF belong, say, $GF(2^L)$. Then the GP coefficients are related by *conjugacy constraints*. Such properties are absent in the case of GPs where the coefficients and function values belong to the same field.

2.4.1 Conjugacy Relations

When Frobenius cycles exist in GPs, the coefficients of GPs satisfy conjugacy constraints. We discuss these constraints under two broad classifications of GSFs, namely, those GSFs whose $k \nmid n$ and those whose $k | n$.

Two possibilities can arise in either case:

- (a) Function values belong to $GF(2^n)$, and not to any of its subfields.
- (b) All the function values belong to $GF(2^n)$ as well as to a subfield of it, say, $GF(2^{n_1})$ (Since the latter is a subfield of the former, $n_1 | n$).

(a) $k \nmid n$

If $k \nmid n$, we note that the field to which the function values belong ($GF(2^n)$ or $GF(2^{n_1})$ as the case may be) is a proper subfield of the field to which the coefficients belong. This is because, since $k \nmid n$, it also does not divide a factor of n (n_1 , in this case), and hence the L.C.M. of k and n as well as the L.C.M. of k and n_1 are respectively not equal to n and n_1 . Therefore, nontrivial conjugacy relations exist among the coefficients

when $k \nmid n$.

We state the following theorem on conjugacy relations:

Theorem 2.4.1: If $k \nmid n$, the coefficients a_i , $i = -\infty, 0, 1, \dots, 2^k-2$, of a GP, satisfy the conjugacy relations given by

$$(a_i)^Q = a_{i \cdot Q \bmod M-1}, \quad i = 0, 1, \dots, 2^k-2 \quad (2.4.2a)$$

and

$$(a_{-\infty})^Q = a_{-\infty}, \quad (2.4.2b)$$

where $M = 2^k$, and Q is equal to

- (1) 2^n , if the function values belong to $GF(2^n)$ and not to any subfield of it.
- (2) 2^{n_1} , if all the function values belong to $GF(2^n)$ as well as to a subfield of it, namely, $GF(2^{n_1})$.

The coefficients belong to $GF(2^L)$, where $L = \text{L.C.M of } k \text{ and, } n \text{ or } n_1 \text{ in (1) and (2) respectively.}$

Proof: When $k \nmid n$, it also does not divide n_1 , a factor of n , and hence the field to which the function values belong, is not equal to the field to which the coefficients belong, and will always be a proper subfield of the latter. Therefore conjugacy relations exist among the coefficients of the GP. This is similar to the conjugacy relations among the DFT coefficients of a sequence over $GF(q)$ of length N , in which case the relation is

$$(A_j)^q = A_{jq \bmod N},$$

where the A_j 's are in an extension field of $GF(q)$.

Since the GT is an extension of the DFT over finite fields, in the former case, we have the DFT of a sequence over $GF(2^n)$ (or $GF(2^{n_1})$) of length 2^k-1 . Hence the conjugacy relations used for DFT can be extended to GTs, to get (2.4.2a). Further (2.4.2b) is valid since $a_{-\infty} = f(\alpha^{-\infty}) \in GF(Q)$. Q.E.D.

For an alternate proof, see [10].

Example 2.4.1: We consider an example of a GSF where $k \nmid n$, and all the function values belong to $GF(2^{n_1})$, a subfield of $GF(2^n)$. Let $n = 6$, $k = 4$ and $n_1 = 3$. Then $L = 12$. Let $x^{12} + x^6 + x^4 + x + 1$ be a primitive polynomial for generating $GF(2^{12})$, with γ as a primitive element. Then the subfield $GF(2^6)$ is generated by the primitive polynomial $x^6 + x^5 + 1$.

Let the function values represented as a power of a primitive element β in $GF(2^6)$ be given by

$$-\infty, 36, 54, 45, 36, 18, 0, 45, 27, 45, 18, 0, 18, 54, 36, 27$$

(where only the exponents of β are listed). It may be noted that all the function values given above are also members of the subfield $GF(2^3)$.

The GSF which realizes this mapping may be found to be (Computational procedures for GSFs are discussed in Section 2.5):

$$\begin{aligned} f(x) = & \gamma^{3510} x^{15} + \gamma^{607} x^{14} + \gamma^{3659} x^{13} + \gamma^{926} x^{12} + \gamma^{1993} x^{11} + \gamma^{3120} x^{10} + \gamma^{3187} x^9 \\ & + \gamma^{3821} x^8 + \gamma^{761} x^7 + \gamma^{3313} x^6 + \gamma^{390} x^5 + \gamma^{1903} x^4 + \gamma^{1934} x^3 + \gamma^{2939} x^2 + \\ & \gamma^{3037} x. \end{aligned}$$

It is easy to verify that the coefficients satisfy conjugacy relations given by

$$(a_i)^8 = a_{8i \bmod 15}, i = 0, 1, \dots, 14.$$

(b) $k|n$

If $k|n$, we note that the fields to which the function values and the coefficients belong, is same if

- (1) the function values belong to $GF(2^n)$, and not to any of its subfields.
- (2) all the function values belong to $GF(2^n)$ as well as to a subfield of it, namely, $GF(2^{n_1})$, and if $k|n_1$.

Thus in cases (1) and (2), the GP coefficients exhibit trivial conjugacy relations and they belong to $GF(2^n)$ and $GF(2^{n_1})$ respectively.

However, conjugacy relations exist if

all the function values $\in \text{GF}(2^n)$ as well as to a subfield of it, namely, $\text{GF}(2^{n_1})$, and if $k \nmid n_1$, in which case the coefficients belong to $\text{GF}(2^L)$, where $L = \text{L.C.M. of } n_1 \text{ and } k$.

Conjugacy relations are same as in Theorem 2.4.1 except that now Q is taken as 2^{n_1} .

Example 2.4.2: In this example, we consider a case where $k|n$, but all the function values belong to a subfield $\text{GF}(2^{n_1})$, where $k \nmid n_1$, and hence nontrivial conjugacy relations exist among the GP coefficients. Let $n = 12, k = 3$ and $n_1 = 4$ so that $k \nmid n_1$. $L = 12$. Let $x^{12} + x^6 + x^4 + x + 1$ be a primitive polynomial for generating $\text{GF}(2^{12})$, with γ as a primitive element. Let the function values be represented as a power of a primitive element γ in $\text{GF}(2^{12})$, the exponents of which are given by

$$-\infty, 273, 1365, 819, 0, 1092, 1638, 546.$$

The above values are also members of $\text{GF}(2^4)$, a subfield of $\text{GF}(2^{12})$.

The GSF which realizes this mapping may be found to be

$$f(x) = \gamma^{1365} x^7 + \gamma^{4066} x^6 + \gamma^{3631} x^5 + \gamma^{3256} x^4 + \gamma^{766} x^3 + \gamma^{2251} x^2 + \gamma^{2956} x.$$

The coefficients satisfy conjugacy relations given by

$$(a_i)^{16} = a_{16i \bmod 7}, i = 0, 1, \dots, 6.$$

2.4.2 Number of Frobenius Cycles

In the following theorem, we give an expression for the number of Frobenius cycles in a single variable GP exhibiting Frobenius properties:

Theorem 2.4.2: The number of Frobenius cycles existing in a single variable GP representing a GSF mapping from $\text{GF}(2^k)$ to $\text{GF}(2^n)$, is given by

$$\text{nfrob} = 1 + \sum_{\substack{D \\ D|M-1}} \phi(D)/\exp_Q(D), \quad (2.4.3)$$

where $M1 = 2^k - 1$, $Q = 2^n$, $\exp_Q D = e$ is the least positive integer such that $Q^e = 1 \pmod{D}$, and $\phi(\cdot)$ is the Euler's phi function.

Proof: Since GT is an extension of DFT over finite fields, expression for the number of conjugate classes (say N_{cdft}) in the case of DFT can be employed with a suitable modification to calculate the number of Frobenius cycles in single variable GPs. The only modification required will be the addition of a '1' to N_{cdft} , to account for the term $a_{-\infty}$. Expression for N_{cdft} can be derived as follows [32]:

Let S be the set $[0, 1, 2, \dots, M1-1]$. Let D be an integer such that $(Q, D) = 1$ (ie., Q is relatively prime to D), $D|M1$, and $\exp_Q(D) = e$ (ie., e is the least positive integer such that $Q^e = 1 \pmod{D}$).

Then $S1 = [1, Q, Q^2, \dots, Q^{e-1}]$ forms a cycle of length ' e '. Now take an integer ℓ_1 such that $\ell_1 < D$, $\ell_1 \notin S1$ and the greatest common divisor (G.C.D.) of ℓ_1 and D (denoted as $(\ell_1, D) = 1$). Using the number theoretic result that if $ax \equiv ay \pmod{m}$ and $(a, m) = d$, then $x \equiv y \pmod{m/d}$, we have $\ell_1 Q^e = \ell_1 \pmod{D}$. Now we form another set $S2 = [\ell_1, \ell_1 Q, \ell_1 Q^2, \dots, \ell_1 Q^{e-1}]$ which is also of length ' e '. It can be proved that $S1$ and $S2$ are disjoint. Because, if this were not so, then we would have $\ell_1 Q^i \equiv Q^j \pmod{D}$, $0 \leq i, j \leq e-1$. As $(Q, D) = 1$, this implies, $\ell_1 \equiv Q^{j-i} \pmod{D}$, which means $\ell_1 \in S1$, which is contradictory to our initial assumption. Hence $S1$ and $S2$ are disjoint.

Next we choose another integer $\ell_2 < D$, such that $\ell_2 \notin S1$, $\ell_2 \notin S2$ and $(\ell_2, D) = 1$, and obtain an $S3$ similar to $S2$, which will be disjoint both with $S1$ and $S2$.

It may be noted that integers which are relatively prime to D form a group under multiplication, whose order is $\phi(D)$, and $S1$ forms a subgroup of this group of order ' e '. Since the order of a subgroup divides the order of its group, we have $e|\phi(D)$, or $\phi(D)/e$ is an integer. Now we have covered $\phi(D)$ elements of S and obtained the number of cycles in this $\phi(D)$ elements as $\phi(D)/e$. The cycles in S corresponding to $S1, S2, \dots$ are $[m', m'Q, \dots, m'Q^{e-1}]$, $[m'\ell_1, m'\ell_1 Q, \dots, m'\ell_1 Q^{e-1}]$, respectively, where $m' = M1/D$. Since from

number theory, we have the result, $\sum_{D|M1} \phi(D) = M1$, we repeat the above steps for other divisors D of $M1$ to exhaust S .

$$\text{Thus Ncdft} = \sum_{\substack{D \\ D|M1}} \phi(D)/\exp_Q(D)$$

and

$$\text{nfrob} = 1 + \text{Ncdft}$$

Q.E.D.

Examples: We illustrate the above theorem with some examples.

Example 2.4.3: Let $n = 1$, and $k = 4$. This is an example of a 4 variable boolean function.

The Frobenius cycles in this case, listed in terms of the GP coefficients are:

- (1) $\{a_{-w}\}$
- (2) $\{a_0\}$
- (3) $\{a_1, a_2, a_4, a_8\}$
- (4) $\{a_3, a_6, a_{12}, a_9\}$
- (5) $\{a_5, a_{10}\}$
- (6) $\{a_7, a_{14}, a_{13}, a_{11}\}$

Using the formula derived, the number of cycles can be calculated as follows:

Here $M1 = 2^4 - 1 = 15$; the divisors (D) of 15 are 1, 3, 5 and 15.

$$Q = 2^1 = 2.$$

By convention, $\exp_Q(1) = 1$,

and $\exp_2(3) = 2$, $\exp_2(5) = 4$, $\exp_2(15) = 4$.

$$\phi(1) = 1, \phi(3) = 2, \phi(5) = 4, \phi(15) = 8.$$

$$\text{Thus nfrob} = 1 + \frac{\phi(1)}{1} + \frac{\phi(3)}{2} + \frac{\phi(5)}{4} + \frac{\phi(15)}{4} = 1 + 1 + 1 + 1 + 2 = 6.$$

Example 2.4.4: Let $n = 2$, and $k = 4$. This is an example of a digital system with 4 input variables and 2 output variables. The Frobenius cycles in this case, listed in terms of the

GP coefficients are:

- (1) $\{a_{-\infty}\}$
- (2) $\{a_0\}$
- (3) $\{a_1, a_4\}$
- (4) $\{a_2, a_8\}$
- (5) $\{a_3, a_{12}\}$
- (6) $\{a_5\}$
- (7) $\{a_6, a_9\}$
- (8) $\{a_7, a_{13}\}$
- (9) $\{a_{10}\}$
- (10) $\{a_{11}, a_{14}\}$

The number of cycles can be calculated using (2.4.3) as follows:

Here $M1 = 2^4 - 1 = 15$; the divisors (D) of 15 are 1, 3, 5 and 15 as in Example 2.4.3.

$$Q = 2^2 = 4.$$

By convention, $\exp_4(1) = 1$,

and $\exp_4(3) = 1$, $\exp_4(5) = 2$, $\exp_4(15) = 2$.

$\phi(1) = 1$, $\phi(3) = 2$, $\phi(5) = 4$, $\phi(15) = 8$ as before.

$$\text{Thus } n_{\text{frob}} = 1 + \frac{\phi(1)}{1} + \frac{\phi(3)}{1} + \frac{\phi(5)}{2} + \frac{\phi(15)}{2} = 1 + 1 + 2 + 2 + 4 = 10.$$

Example 2.4.5: Let us consider $n = 4$, and $k = 4$. This is an example of a mapping which has a domain and a range assuming values from the same finite extension field, such as a permuter. We illustrate the fact that conjugacy relations among the GP coefficients are trivial in this case and all the coefficients are independent of each other.

Using (2.4.3), the number of cycles is calculated as follows:

Here $M1 = 2^4 - 1 = 15$; the divisors (D) of 15 are 1, 3, 5 and 15.

$$Q = 2^4 = 16.$$

By	110030
and \exp_{16}	117849
Thus nfro	125691
Hei	143418
independe	143493

2.4.3

Con
sum of var
algebra. A
direct sum
by assigning
of the Frobenius
classes to zero
number of n
as given by
algebra may

2.4.4

A GSI
the terms in
represented a
Frobenius poly

From the discussion in Section 2.4.3, we may say that any FF may be represented as a sum of elements of minimal ideals in the corresponding monoid algebra.

2.5 Computational Procedures

In this section, we discuss various procedures for computation of GP coefficients and function values of a GSF mapping from $GF(2^k)$ to $GF(2^n)$.

2.5.1 Computation of Coefficients

We saw that the Galois Transform matrix \mathcal{G} of size $2^k \times 2^k$, is an extension of the corresponding DFT matrix of size $2^{k-1} \times 2^{k-1}$. Hence the problem of computation of the GP coefficients reduces to the problem of computing the corresponding DFT coefficients of a sequence of function values $f(\alpha^i)$, $i = 0, 1, \dots, 2^{k-2}$, over $GF(2^n)$ of length 2^{k-1} , and then adding the coefficient $a_{-\omega} = f(\alpha^{-\omega})$ to the first coefficient, $(a_0 - a_{-\omega})$, in the resulting DFT sequence. Various standard algorithms collectively known as *fast Fourier transform (FFT) algorithms*, are available for the fast computation of DFT for different data lengths [20], and the same can be employed for computing the GP coefficients.

2.5.2 Computation of Function Values

Given the coefficients of a GP, several methods are available to compute the values of the corresponding function. The method used depends on the requirement and the type of the GP under consideration.

If the requirement is to compute all the function values, then an obvious method would be to compute the inverse DFT of the sequence $[(a_0 - a_{-\omega}), a_1, a_2, \dots, a_\xi]$ where $\xi = 2^{k-2}$, to get $f(\alpha^i)$, $i = 0, 1, \dots, 2^{k-2}$, using FFT techniques, and then put $f(\alpha^{-\omega}) = a_{-\omega}$.

If the requirement is to compute only some values, then we can use any of the following two techniques depending on whether Frobenius cycles exist or not.

2.5.2.1 Frobenius Sum Computing

This method can be used only for those GPs which may be expressed as a Frobenius polynomial (FP). Therefore the function value computation reduces to Frobenius sum

computing. A simple procedure exists for computing the values of a FP, say $f(x)$. Let us assume for the present that the constant term is 0. Then let $f(x) = \sum_j a_j x^j$, where $a_j \in GF(2^L)$, and j is a member of the conjugacy class $\{j, jQ, jQ^2, \dots, jQ^{t-1}\}$ where $jQ^{2^t} = j$ modulo $2^k - 1$, and $Q = 2^n$.

The value of $f(x) = y_{n-1} \dots y_1 y_0$, at any $x = x_{k-1} \dots x_1 x_0$, can be computed as follows:

- (1) Express $x_{k-1} \dots x_1 x_0$ as a power of the primitive element α in $GF(2^k)$ say, α^i [ie., convert x from cartesian form to polar form].
- (2) Express α^i in terms of the primitive element γ in $GF(2^L)$; $\alpha^i = \gamma^{\nu \cdot i} = \gamma^t$ (say), since $\alpha = \gamma^\nu$, where $\nu = 2^L - 1 / 2^k - 1$.
- (3) Find the arguments of each Frobenius term, ie., $a_j x^j = a_j \gamma^{t \cdot j} = \gamma^{r_j}$ (say).
- (4) Compute each of the Frobenius sums $\text{frs}(\gamma^{r_j})$ and take their sum to get $y = y_{n-1} \dots y_1 y_0$.

If we work in *normal basis* (NB), the computation of Frobenius sum becomes very easy, as squaring of a field element can be done in NB by a mere cyclic shift.

Example 2.5.1: We illustrate the computations with GPs in this example. For this purpose, we use the mapping which was considered in Section 2.1, with its truth table given in Table 2.3. Here $k \nmid n$ and thus conjugacy relations exist among the GP coefficients. Let $n = 2$ and $k = 3$. Then the GP coefficients lie in $GF(2^6)$ since $L = \text{L.C.M. of } 2 \text{ and } 3 = 6$. Let γ be a primitive element of $GF(2^6)$ and let its minimal polynomial over $GF(2)$ be $x^6 + x + 1$. We choose this polynomial for generating the finite field $GF(2^6)$.

Let $\alpha =$ primitive element of $GF(2^3)$ and $\beta =$ primitive element of $GF(2^2)$.

Then $\alpha = \gamma^{2^6 - 1 / 2^3 - 1} = \gamma^9$ and $\beta = \gamma^{2^6 - 1 / 2^2 - 1} = \gamma^{21}$.

Minimal polynomial of α over $GF(2) = (x - \gamma^9)(x - \gamma^{18})(x - \gamma^{36}) = x^3 + x^2 + 1$.

Minimal polynomial of β over $GF(2) = (x - \gamma^{21})(x - \gamma^{42}) = x^2 + x + 1$.

Thus the subfields $GF(2^3)$ and $GF(2^2)$ are generated by the minimal polynomials of γ^9 and γ^{21} respectively.

The GP coefficients representing this mapping can be calculated as

$$a_{-\infty} = 0, a_0 = 0, a_1 = \gamma^{32}, a_2 = \gamma^8, a_3 = \gamma^{28}, a_4 = \gamma^2, a_5 = \gamma^{49}, a_6 = \gamma^7.$$

$$\begin{aligned} \text{Thus } f(x) &= \gamma^{32} x^6 + \gamma^8 x^5 + \gamma^{28} x^4 + \gamma^2 x^3 + \gamma^{49} x^2 + \gamma^7 x \\ &= \text{frs}(\gamma^7 x) + \text{frs}(\gamma^2 x^3), \end{aligned}$$

(since $f(x)$ is composed of two Frobenius cycles).

It may be noted that

$$\begin{aligned} \text{frs}(\gamma^7 x) &= (\gamma^7 x) + (\gamma^7 x)^4 + (\gamma^7 x)^{16} = \gamma^7 x + \gamma^{28} x^4 + \gamma^{49} x^2 \quad \text{and} \\ \text{frs}(\gamma^2 x^3) &= (\gamma^2 x^3) + (\gamma^2 x^3)^4 + (\gamma^2 x^3)^{16} = \gamma^2 x^3 + \gamma^8 x^5 + \gamma^{32} x^6. \end{aligned}$$

Now let us compute the value of $f(x)$ at, say $x = x_2 x_1 x_0 = 0 1 1$.

$x_2 x_1 x_0$ expressed as a power of α in $GF(2^3)$, from Table 2.3, is α^5 .

α^5 expressed in terms of the primitive element γ in $GF(2^L) = \gamma^{9 \times 5} = \gamma^{45}$.

The two arguments of the Frobenius function are obtained as $\gamma^7 x = \gamma^7 \cdot \gamma^{45} = \gamma^{52}$ and $\gamma^2 x^3 = \gamma^2 \cdot \gamma^{3 \times 45} = \gamma^{11}$.

Computing the Frobenius sums $\text{frs}(\gamma^{52})$ and $\text{frs}(\gamma^{11})$, we get

$$\text{frs}(\gamma^{52}) = \gamma^{52} + \gamma^{19} + \gamma^{13} = \gamma^0 = \beta^0 = 1, \quad \text{and}$$

$$\text{frs}(\gamma^{11}) = \gamma^{11} + \gamma^{44} + \gamma^{50} = \gamma^{42} = \beta^2.$$

$$\text{Thus } f(0 1 1) = f(\alpha^5) = \beta^0 + \beta^2 = \beta = 1 0.$$

It may be verified from the truth table (Table 2.3) that the above is true.

2.5.2.2 Horner's Rule

The Horner's rule is a standard method for polynomial computation, irrespective of the type of GP under consideration. To describe the method of computation, let us consider the GP:

$$f(x) = a_{-\infty} + a_0 x^{2^k-1} + a_1 x^{2^k-2} + \dots + a_i x^{2^k-1-i} + \dots + a_\xi x,$$

where $\xi = 2^k - 2$.

This may also be written as

$$f(x) = ((((((a_0 x + a_1) x + a_2) x + a_3) x + \dots + a_{\xi-1}) x + a_\xi) x + a_{-\infty}). \quad (2.5.1)$$

(2.5.1) suggests a recursive procedure for polynomial computation and is illustrated in Figure 2.1.

The Horner's Polynomial Computer consists of a finite field multiplier and a finite field adder, working, in general, in $GF(2^L)$, L being the L.C.M. of n and k . Initially, assume that the multiplier output is zero, so that a_0 is available at one of the inputs to the multiplier, the other input being the value $x \in GF(2^k)$ represented in $GF(2^L)$, at which it is required to compute the function. Thus, at the arrival of the first clock pulse, a_0 is multiplied by x and a_1 gets added to it, and becomes available at the input to the multiplier before the arrival of the second clock pulse, and so on. Lastly, after the final multiplication, $a_{-\infty}$ is added, to get the value of the function at x , at the adder output. It takes $2^k - 1$ multiplications and additions to compute one function value.

Example 2.5.2: We take the same function as in Example 2.5.1 to illustrate Horner's polynomial computation.

$$\text{We had } f(x) = \gamma^{32} x^6 + \gamma^8 x^5 + \gamma^{28} x^4 + \gamma^2 x^3 + \gamma^{49} x^2 + \gamma^7 x.$$

$$\text{ie., } a_{-\infty} = 0, a_0 = 0, a_1 = \gamma^{32}, a_2 = \gamma^8, a_3 = \gamma^{28}, a_4 = \gamma^2, a_5 = \gamma^{49}, a_6 = \gamma^7.$$

$$\text{Now let us compute the value of } f(x) \text{ at } x = x_2 x_1 x_0 = 011 = \alpha^5 = \gamma^{9 \times 5} = \gamma^{45}.$$

$$\begin{aligned} f(x) &= (((((((0 \cdot \gamma^{45} + \gamma^{32}) \gamma^{45} + \gamma^8) \gamma^{45} + \gamma^{28}) \gamma^{45} + \gamma^2) \gamma^{45} + \gamma^{49}) \gamma^{45} + \gamma^7) \gamma^{45} + 0 \\ &= \gamma^{21} = \beta = 10. \end{aligned}$$

be seen from the truth table given in Table 2.3 that the above is true.

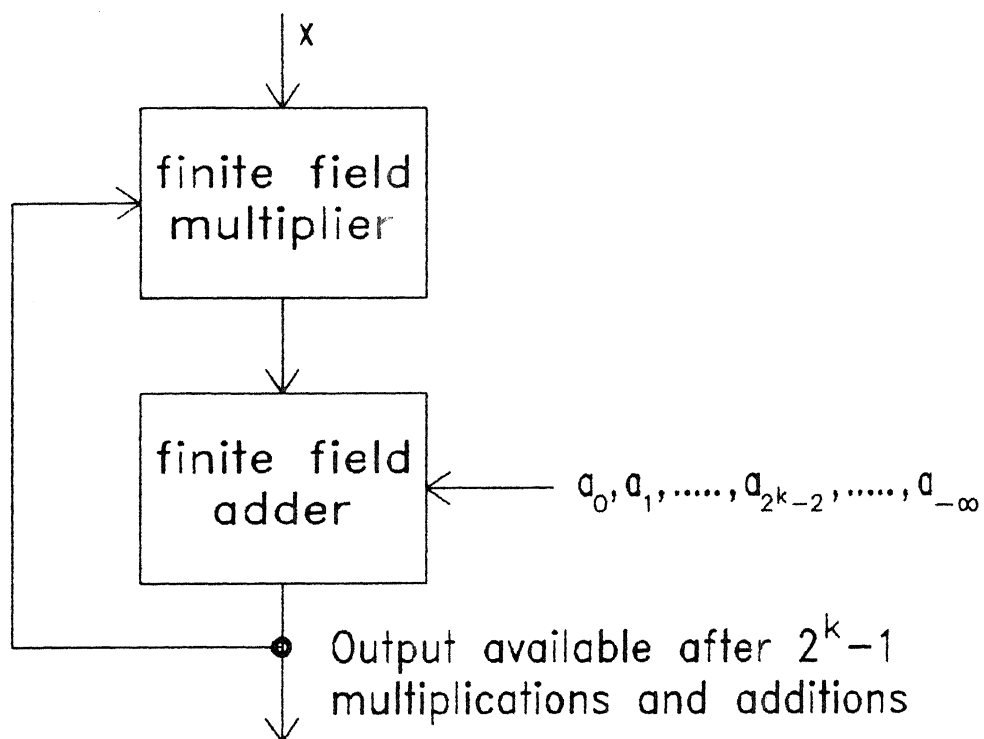


Fig 2.1: Horner's Polynomial Computer

2.6 Multi-Dimensional Galois Switching Functions

Although our studies are confined to 1-D and 2-D GSFs, we state a theorem on multi-dimensional GSFs and prove it for the 2-D case:

Theorem 2.6.1: Any m -dimensional GSF $f(x_1, x_2, \dots, x_m)$ can be represented by an m -variable GP

$$f(x_1, x_2, \dots, x_m) = \sum_{j_1} \sum_{j_2} \dots \sum_{j_m} a_{j_1 j_2 \dots j_m} x_1^{-j_1} x_2^{-j_2} \dots x_m^{-j_m},$$

$$j_i = -\infty, 0, 1, \dots, 2^{k_i} - 2, \quad i = 1, 2, \dots, m, \quad \text{where } -j_i \text{ is taken modulo } 2^{k_i} - 1 \quad (2.6.1)$$

The coefficients are given by

$$a_{-\infty -\infty \dots -\infty} = f(\alpha_1^{-\infty}, \alpha_2^{-\infty}, \dots, \alpha_m^{-\infty}) \quad (2.6.2a)$$

$$a_{j_1 -\infty -\infty \dots -\infty} = \sum_{x_1} x_1^{j_1} f(x_1, \alpha_2^{-\infty}, \alpha_3^{-\infty}, \dots, \alpha_m^{-\infty})$$

$$a_{-\infty j_2 -\infty \dots -\infty} = \sum_{x_2} x_2^{j_2} f(\alpha_1^{-\infty}, x_2, \alpha_3^{-\infty}, \dots, \alpha_m^{-\infty})$$

⋮

${}^m C_1$ coefficients

$$a_{-\infty -\infty -\infty \dots j_m} = \sum_{x_m} x_m^{j_m} f(\alpha_1^{-\infty}, \alpha_2^{-\infty}, \alpha_3^{-\infty}, \dots, x_m)$$

(2.6.2b)

$$\begin{aligned}
 a_{j_1 j_2 -\infty \dots -\infty} &= \sum_{x_1} \sum_{x_2} x_1^{j_1} x_2^{j_2} f(x_1, x_2, \alpha_3^{-\infty}, \dots, \alpha_m^{-\infty}) \\
 a_{j_1 -\infty j_3 \dots -\infty} &= \sum_{x_1} \sum_{x_3} x_1^{j_1} x_3^{j_3} f(x_1, \alpha_2^{-\infty}, x_3, \dots, \alpha_m^{-\infty}) \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 a_{-\infty -\infty \dots j_{m-1} j_m} &= \sum_{x_{m-1}} \sum_{x_m} x_{m-1}^{j_{m-1}} x_m^{j_m} f(\alpha_1^{-\infty}, \alpha_2^{-\infty}, \dots, x_{m-1}, x_m) \\
 &\vdots \\
 &\vdots \\
 &\vdots
 \end{aligned}
 \tag{2.6.2c}$$

$$a_{j_1 j_2 \dots j_m} = \sum_{x_1} \sum_{x_2} \dots \sum_{x_m} x_1^{j_1} x_2^{j_2} \dots x_m^{j_m} f(x_1, x_2, \dots, x_m), \tag{2.6.2d}$$

$j_i = 0, 1, \dots, 2^{k_i} - 2$, $x_i \in GF(2^{k_i})$, α_i is a primitive element of $GF(2^{k_i})$, $i = 1, 2, \dots, m$.
 $f(x_1, x_2, \dots, x_m) \in GF(2^n)$, and the coefficients (a's) $\in GF(2^L)$, L being the L.C.M. of n and k_i , $i = 1, 2, \dots, m$.

Note: It may be noted that in (2.6.1), any $x_i^{-\infty}$, $i = 1, 2, \dots, m$, is taken as 1. Thus for $j_i = -\infty$, $i = 1, 2, \dots, m$, we get
 $f(\alpha_1^{-\infty}, \alpha_2^{-\infty}, \dots, \alpha_m^{-\infty}) = a_{-\infty -\infty \dots -\infty}$ = the constant term.

For proof, it is sufficient to consider the two-variable case, since it may then be formally extended to the m variable case.

2.6.1 Two-Dimensional GSFs

The theorem may be reformulated for two variable GPs as follows:

Theorem 2.6.2: Any 2-D GSF, $f(x_1, x_2)$, can be represented by a 2 variable GP

$$f(x_1, x_2) = \sum_{j_1} \sum_{j_2} a_{j_1 j_2} x_1^{-j_1} x_2^{-j_2}, \quad (2.6.3)$$

$j_i = -\infty, 0, 1, \dots, 2^{k_i} - 2$, $i = 1, 2$, where $-j_i$ is taken modulo $2^{k_i} - 1$.

[assuming that $(x_i)^{-\infty}$ is taken as 1 for $i = 1, 2$].

The coefficients are given by

$$a_{-\infty -\infty} = f(\alpha_1^{-\infty}, \alpha_2^{-\infty}), \quad (2.6.4a)$$

$$\left. \begin{aligned} a_{j_1 -\infty} &= \sum_{x_1} x_1^{j_1} f(x_1, \alpha_2^{-\infty}) \\ &\vdots \\ {}^2C_1 &= 2 \text{ coefficients} \\ a_{-\infty j_2} &= \sum_{x_2} x_2^{j_2} f(\alpha_1^{-\infty}, x_2) \end{aligned} \right\} \quad (2.6.4b)$$

$$a_{j_1 j_2} = \sum_{x_1} \sum_{x_2} x_1^{j_1} x_2^{j_2} f(x_1, x_2), \quad (2.6.4c)$$

$j_i = 0, 1, \dots, 2^{k_i} - 2$, $x_i \in \text{GF}(2^{k_i})$, α_i is a primitive element of $\text{GF}(2^{k_i})$, $i = 1, 2$,

$f(x_1, x_2) \in \text{GF}(2^L)$, and the coefficients (a's) $\in \text{GF}(2^L)$, L being the L.C.M. of n and k_i ,

$i = 1, 2$.

Proof: With the assumption that when $j_1, j_2 = -\infty$, $x_1^{j_1}$ and $x_2^{j_2}$ becomes unity, whereas the argument x_1 and x_2 of $f(x_1, x_2)$ becomes $0 = \alpha^{-\infty}$, we multiply $f(x_1, x_2)$ by $x_1^{j_1} x_2^{j_2}$ on both sides and then sum over all $x_1 \in \text{GF}(2^{k_1})$ and $x_2 \in \text{GF}(2^{k_2})$ where $j_i = -\infty, 0, 1, \dots, 2^{k_i} - 2$, $i = 1, 2$. Summation over all $x_1 \in \text{GF}(2^{k_1})$ and $x_2 \in \text{GF}(2^{k_2})$ forces all the terms on

the right hand side of the expression to zero except the term corresponding to j_i , $i = 1, 2$, which becomes $a_{j_1 j_2} \sum_{x_1} \sum_{x_2} x_1^{2^{k_1}-1} x_2^{2^{k_2}-1}$. Since in any finite field $GF(2^{k_i})$, $x_i^{2^{k_i}-1} = 1$, the summation reduces to unity. Thus we get,

$$\sum_{x_1} \sum_{x_2} x_1^{j_1} x_2^{j_2} f(x_1, x_2) = a_{j_1 j_2}$$

The fact that the coefficients belong to $GF(2^L)$, L being the L.C.M. of n and k_i , $i = 1, 2$, may be proved by noting the following:

If we arrange the function values $f(x_1, x_2)$, $x_i \in GF(2^{k_i})$, $i = 1, 2$, in the form of a $2^{k_1} \times 2^{k_2}$ matrix, then the coefficients of the 2 variable GP can be obtained by

- (1) computing the 1-D Galois transform (GT) coefficients of the rows of the matrix, which represent a mapping from $GF(2^{k_1})$ to $GF(2^{L_1})$, and replacing the rows with the resulting coefficients which belong to say, $GF(2^{L_1})$, L_1 being the L.C.M. of n and k_1 , followed by
- (2) computing the 1-D GT coefficients of the resulting columns, which now represent a mapping from $GF(2^{k_2})$ to $GF(2^{L_1})$. The resulting final coefficients belong to $GF(2^L)$, L being the L.C.M. of k_2 and L_1 , which is the same as the L.C.M. of k_1 , k_2 and n .

[The computation of the coefficients can also be done by taking the columns first and then proceeding to the rows.]

2.6.1.1 Conjugacy Relations

The conjugacy relations for single variable GPs can be extended to the two-variable case, and is stated below without proof:

Theorem 2.6.3: If at least one $k_i \nmid n$, the coefficients $a_{j_1 j_2}$, $j_i = 0, 1, \dots, 2^{k_i}-2$,

$i = 1, 2$, of the two variable GP defined in (2.6.3), satisfy the conjugacy relations given by

$$(a_{j_1 j_2})^Q = a_{j_1}^{Q \pmod{2^{k_1}-1}} a_{j_2}^{Q \pmod{2^{k_2}-1}},$$

$$j_i = -\infty, 0, 1, \dots, 2^{k_i}-2. \quad (2.6.5)$$

with the assumption that when $j_i = -\infty$, $i = 1, 2$, $\alpha_i^{-\infty \pmod{2^{k_i}-1}}$ is taken as $\alpha_i^{-\infty}$, where $Q = 2^n$.

Note: Situations arising due to all the function values belonging to a subfield $GF(2^{n_1})$ of $GF(2^n)$, in cases of at least one $k_i \nmid n$, and $k_i | n$ for all i , are similar to the single variable case and hence we do not consider them here.

2.6.1.2 Number of Frobenius Cycles

We extend the result on the number of Frobenius cycles in single variable GPs to the two-variable case. The theorem is stated as follows:

Theorem 2.6.4: The number of Frobenius cycles in the case of two-variable GPs, is given by

$$\begin{aligned} \text{nfrob2D} = 1 + & \sum_{\substack{D_1 \\ D_1 | M_1}} \phi(D_1) / \exp_Q(D_1) + \sum_{\substack{D_2 \\ D_2 | M_2}} \phi(D_2) / \exp_Q(D_2) + \\ & \sum_{\substack{D_1 \\ D_1 | M_1}} \sum_{\substack{D_2 \\ D_2 | M_2}} \phi(D_1) \phi(D_2) / \text{L.C.M}(\exp_Q(D_1), \exp_Q(D_2)). \end{aligned} \quad (2.6.6)$$

Proof: The term '1' accounts for the coefficient $a_{-\infty -\infty}$. The second and third terms account for the number of 1-D Frobenius cycles (Frobenius cycles corresponding to the single variable GPs of the first row and first column). The last term accounts for the number of 2-D Frobenius cycles formed by taking pairwise products of the 1-D Frobenius cycles, and may be obtained as follows [32]:

Consider a 1-D Frobenius cycle corresponding to a divisor D_1 of M_1 . We know from

the single variable case that there are $\phi(D_1)/\ell_1$ such cycles each of length ℓ_1 , where $\ell_1 = \exp_Q(D_1)$. Similarly, there are $\phi(D_2)/\ell_2$, 1-D cycles each of length ℓ_2 , where $\ell_2 = \exp_Q(D_2)$, corresponding to a divisor D_2 of M_2 . Therefore the number of product terms formed by taking pairwise products of the 1-D cycles corresponding to a divisor D_1 of M_1 and a divisor D_2 of M_2 is equal to $\phi(D_1)\phi(D_2)/\ell_1\ell_2$. Now we count the number of 2-D Frobenius cycles in each product. It is known that the length of such a 2-D cycle is equal to the L.C.M. of ℓ_1 and ℓ_2 . Thus the number of 2-D cycles in each product is equal to $\ell_1\ell_2/\text{L.C.M.}(\ell_1, \ell_2)$ (which is equal to the G.C.D. of ℓ_1 and ℓ_2). Therefore the total number of 2-D Frobenius cycles corresponding to D_1 and D_2 is equal to $\phi(D_1)\phi(D_2)/\text{L.C.M.}(\ell_1, \ell_2)$. Summing over all D_1 and D_2 , we obtain the third term in the expression for nfrob2D . Q.E.D.

Examples

Example 2.6.1: Let $n = 2$, $k_1 = 3$, $k_2 = 2$.

Then $M_1 = 2^{k_1} - 1 = 7$; $M_2 = 2^{k_2} - 1 = 3$; $Q = 2^n = 4$.

The divisors of $7 = D_1 = 1, 7$.

The divisors of $3 = D_2 = 1, 3$.

$$\exp_4(1) = 1; \exp_4(7) = 3; \exp_4(3) = 1.$$

$$\begin{aligned} \text{Thus nfrob2D} &= 1 + \sum_{\substack{D_1 \\ D_1 | 7}} \phi(D_1)/\exp_4(D_1) + \sum_{\substack{D_2 \\ D_2 | 3}} \phi(D_2)/\exp_4(D_2) + \\ &\quad \sum_{\substack{D_1 \\ D_1 | 7}} \sum_{\substack{D_2 \\ D_2 | 3}} \phi(D_1)\phi(D_2)/\text{L.C.M.}(\exp_4(D_1), \exp_4(D_2)) \\ &= 1 + \phi(1)/\exp_4(1) + \phi(7)/\exp_4(7) + \phi(1)/\exp_4(1) + \phi(3)/\exp_4(3) \\ &\quad + \phi(1) \cdot \phi(1)/\text{L.C.M.}(\exp_4(1), \exp_4(1)) + \phi(1) \cdot \phi(3)/\text{L.C.M.}(\exp_4(1), \\ &\quad \exp_4(3)) + \phi(7) \cdot \phi(1)/\text{L.C.M.}(\exp_4(7), \exp_4(1)) + \\ &\quad \phi(7) \cdot \phi(3)/\text{L.C.M.}(\exp_4(7), \exp_4(3)) \end{aligned}$$

$$= 1 + 1 + 2 + 1 + 2 + 1 + 2 + 2 + 4 = 16.$$

Now we list these conjugacy classes to verify the above number:

In each class, the exponents j_1 and j_2 of $a_{j_1 j_2}$, are listed.

The conjugacy relations in this case, is

$$(a_{j_1 j_2})^4 = a_{4.j_1 \pmod{7} \ 4.j_2 \pmod{3}}'$$

$$j_1 = -\infty, 0, 1, \dots, 6, \quad j_2 = -\infty, 0, 1, 2.$$

Now we list the conjugacy classes to verify their number obtained by (2.6.6): (In each class, the exponents j_1 and j_2 of $a_{j_1 j_2}$, are listed.)

- | | | | |
|--|------------------------------------|-------------------------|-------------------------|
| (1) $\{(-\infty, -\infty)\},$ | (2) $\{(-\infty, 0)\},$ | (3) $\{(-\infty, 1)\},$ | (4) $\{(-\infty, 2)\},$ |
| (5) $\{(0, -\infty)\},$ | (6) $\{(0, 0)\},$ | (7) $\{(0, 1)\},$ | (8) $\{(0, 2)\},$ |
| (9) $\{(1, -\infty), (4, -\infty), (2, -\infty)\},$ | (10) $\{(1, 0), (4, 0), (2, 0)\},$ | | |
| (11) $\{(1, 1), (4, 1), (2, 1)\},$ | (12) $\{(1, 2), (4, 2), (2, 2)\},$ | | |
| (13) $\{(3, -\infty), (5, -\infty), (6, -\infty)\},$ | (14) $\{(3, 0), (5, 0), (6, 0)\},$ | | |
| (15) $\{(3, 1), (5, 1), (6, 1)\},$ | (16) $\{(3, 2), (5, 2), (6, 2)\}.$ | | |

We consider one more example before concluding this chapter.

Example 2.6.2: Let $n = 1$, $k_1 = 3$, $k_2 = 4$.

Then $M_1 = 2^{k_1} - 1 = 7$; $M_2 = 2^{k_2} - 1 = 15$; $Q = 2$.

The divisors of $7 = D_1 = 1, 7$.

The divisors of $15 = D_2 = 1, 3, 5, 15$.

$$\exp_2(1) = 1; \exp_2(7) = 3; \exp_2(3) = 2, \exp_2(5) = 4; \exp_2(15) = 4.$$

$$\text{Now } 1 + \sum_{\substack{D_1 \\ D_1 | 7}} \phi(D_1)/\exp_2(D_1) = 1 + \phi(1)/\exp_2(1) + \phi(7)/\exp_2(7) = 1 + 1 + 2 = 4.$$

Similarly,

$$\sum_{\substack{D_2 \\ D_2 | 15}} \phi(D_2)/\exp_2(D_2) = \phi(1)/\exp_2(1) + \phi(3)/\exp_2(3) + \phi(5)/\exp_2(5) + \phi(15)/\exp_2(15) \\ = 1 + 1 + 1 + 2 = 5.$$

$$\text{Lastly } \sum_{\substack{D_1 \\ D_1 | 7}} \sum_{\substack{D_2 \\ D_2 | 15}} \phi(D_1)\phi(D_2)/\text{L.C.M}(\exp_2(D_1), \exp_2(D_2)) \\ = \phi(1).\phi(1)/\text{L.C.M}(\exp_2(1), \exp_2(1)) + \phi(1).\phi(3)/\text{L.C.M}(\exp_2(1), \exp_2(3)) + \\ \phi(1).\phi(5)/\text{L.C.M}(\exp_2(1), \exp_2(5)) + \phi(1).\phi(15)/\text{L.C.M}(\exp_2(1), \exp_2(15)) + \\ \phi(7).\phi(1)/\text{L.C.M}(\exp_2(7), \exp_2(1)) + \phi(7).\phi(3)/\text{L.C.M}(\exp_2(7), \exp_2(3)) + \\ \phi(7).\phi(5)/\text{L.C.M}(\exp_2(7), \exp_2(5)) + \phi(7).\phi(15)/\text{L.C.M}(\exp_2(7), \exp_2(15)) \\ = 1 + 1 + 1 + 2 + 2 + 2 + 2 + 4 = 15.$$

$$\text{Thus } \text{nfrob}2D = 4 + 5 + 15 = 24.$$

Now we list these conjugacy classes. As in Example 2.6.1, the exponents j_1 and j_2 of $a_{j_1 j_2}$, are listed in each class:

The conjugacy relations in this case, are

$$(a_{j_1 j_2})^2 = a_{2.j_1 \pmod{7} \ 2.j_2 \pmod{15}},$$

$$j_1 = -\infty, 0, 1, \dots, 6, j_2 = -\infty, 0, 1, \dots, 14.$$

$$(1) \{(-\infty, -\infty)\}$$

$$(2) \{(-\infty, 0)\}$$

$$(3) \{(-\infty, 1), (-\infty, 2), (-\infty, 4), (-\infty, 8)\}$$

$$(4) \{(-\infty, 3), (-\infty, 6), (-\infty, 12), (-\infty, 9)\}$$

$$(5) \{(-\infty, 5), (-\infty, 10)\}$$

$$(6) \{(-\infty, 7), (-\infty, 14), (-\infty, 13), (-\infty, 11)\}$$

$$(7) \{(0, -\infty)\}$$

CENTRAL LIBRARY
U. T. CANCER
Acc. No. A. 114020

- 1) $\{(1, -\infty), (2, -\infty), (4, -\infty)\}$
 2) $\{(0, 3), (0, 6), (0, 12), (0, 9)\}$
 3) $\{(0, 7), (0, 14), (0, 13), (0, 11)\}$
 4) $\{(1, 1), (2, 2), (4, 4), (1, 8), (2, 1), (4, 2), (1, 4), (2, 8), (4, 1), (1, 2), (2, 4), (4, 8)\}$
 5) $\{(1, 3), (2, 6), (4, 12), (1, 9), (2, 3), (4, 6), (1, 12), (2, 9), (4, 3), (1, 6), (2, 12), (4, 9)\}$
 6) $\{(1, 5), (2, 10), (4, 5), (1, 10), (2, 5), (4, 10)\}$
 7) $\{(1, 7), (2, 14), (4, 13), (1, 11), (2, 7), (4, 14), (1, 13), (2, 11), (4, 7), (1, 14), (2, 13), (4, 11)\}$
 8) $\{(2, 0), (4, 0), (1, 0)\}$
 9) $\{(3, 0), (6, 0), (5, 0)\}$
 10) $\{(3, 1), (6, 2), (5, 4), (3, 8), (6, 1), (5, 2), (3, 4), (6, 8), (5, 1), (3, 2), (6, 4), (5, 8)\}$
 11) $\{(3, 3), (6, 6), (5, 12), (3, 9), (6, 3), (5, 6), (3, 12), (6, 9), (5, 3), (3, 6), (6, 12), (5, 9)\}$
 12) $\{(3, 5), (6, 10), (5, 5), (3, 10), (6, 5), (5, 10)\}$
 13) $\{(3, 7), (6, 14), (5, 13), (3, 11), (6, 7), (5, 14), (3, 13), (6, 11), (5, 7), (3, 14), (6, 13), (5, 11)\}$
 14) $\{(3, -\infty), (6, -\infty), (5, -\infty)\}$
 15) $\{(0, 1), (0, 2), (0, 4), (0, 8)\}$
 16) $\{(0, 5), (0, 10)\}$

CHAPTER 3

LINEARIZED GALOIS SWITCHING FUNCTIONS

In this chapter, we study *linearized GSFs* (LGSFs), a class of GSFs representing linear mappings from $GF(2^k)$ to $GF(2^n)$ and represented by *linearized Galois polynomials* (LGP). LGSFs are shown to constitute an *ideal* in the monoid algebra of GSFs. We establish a one-to-one correspondence between LGSFs and *linear (n,k) transformations* (linear mappings from $GF(2^k)$ to $GF(2^n)$), where k is not necessarily equal to n . Depending on whether $k|n$ or $k \nmid n$, the class of LGSFs is broadly divided into two subclasses, and their algebraic structures are studied.

3.1 Conditions for GSFs to be Linear

We first derive the conditions for GSFs to be linear. A function $f(x)$ is said to be linear, if $f(x_1+x_2) = f(x_1) + f(x_2)$. Now, any GSF mapping from $GF(2^k)$ to $GF(2^n)$ described by a GP with coefficients from $GF(2^L)$, L being the L.C.M. of n and k , has been represented in (2.3.1) as

$$f(x) = a_{-\infty} + \sum_{j=0}^{\xi} a_j x^{-j} ; \xi = 2^k - 2,$$

where $-j$ is taken modulo $2^k - 1$ and the α 's in the indices of a 's in (2.3.1) has been omitted.

$$\text{At } x = x_1, f(x) = f(x_1) = a_{-\infty} + \sum_{j=0}^{\xi} a_j x_1^{-j}.$$

$$\text{At } x = x_2, f(x) = f(x_2) = a_{-\infty} + \sum_{j=0}^{\zeta} a_j x_2^{-j}.$$

$$\text{At } x = x_1 + x_2, f(x) = f(x_1 + x_2) = a_{-\infty} + \sum_{j=0}^{\zeta} a_j (x_1 + x_2)^{-j}. \quad (3.1.1)$$

$$\text{Now, } f(x_1) + f(x_2) = \sum_{j=0}^{\zeta} a_j (x_1^{-j} + x_2^{-j}). \quad (3.1.2)$$

For function linearity, we require that (3.1.1) and (3.1.2) be equal. At this point, we use the well known result that, for any field of characteristic p , $(x_1 + x_2)^{p^i} = x_1^{p^i} + x_2^{p^i}$. In our case, we have $p = 2$, and the result simply means that squaring is a linear operation in fields of characteristic 2. Equating (3.1.1) and (3.1.2), we see that, for $f(x)$ to be linear, the only nonzero coefficients can be those corresponding to $j = -2^i$, $i = 0, 1, 2, \dots, k-1$. All the remaining coefficients including the constant term $a_{-\infty}$ must be equal to zero.

We call GSFs with the above constraints by the term *linearized GSFs (LGSFs)*, and the corresponding GPs representing them by the term *linearized Galois polynomials (LGPs)*.

Definition 3.1.1: A LGSF is a GSF described by a LGP of the form

$$f(x) = \sum_{i=0}^{k-1} a_{-2^i} x^{2^i}, \quad (3.1.3)$$

where -2^i is taken modulo $2^k - 1$.

Note: A LGSF is a signal vector of length 2^k over $GF(2^n)$. Therefore it can be described in terms of its *Galois spectrum (Galois transform (GT) coefficients)* as follows:

A LGSF is a signal vector of length 2^k over $GF(2^n)$ whose *Galois spectrum* is identically equal to zero except in those indices -2^i , $i = 0, 1, \dots, k-1$, where -2^i is taken modulo $2^k - 1$.

3.2 Linearized Frobenius Functions

We have seen that the coefficients of any GP representing a mapping from $GF(2^k)$ to $GF(2^n)$ satisfy nontrivial conjugacy relations if the field to which the function values belong (which is $GF(2^n)$ or a subfield of it) is a proper subfield of the field to which the coefficients belong. We called such functions as *Frobenius functions (FFs)* and the associated polynomials as *Frobenius polynomials (FPs)*. This property is naturally carried over to LGSFs representing linear mappings of a similar nature. We call such LGSFs as *linearized Frobenius functions (LFFs)* and the LGPs representing them as *linearized Frobenius polynomials (LFPs)*.

Definition 3.2.1: A LFF is a LGSF representing a linear mapping from $GF(2^k)$ to $GF(2^n)$, and described by a LFP, where a LFP is an LGP whose coefficients satisfy nontrivial conjugacy relations and hence the linearized terms can be grouped into Frobenius cycles.

Note: A LFF is a LGSF whose LGP representation satisfy nontrivial conjugacy relations among its coefficients. It can be described in terms of its Galois spectrum as follows:

A LFF is a signal vector of length 2^k over $GF(2^n)$ whose *Galois spectrum* is identically equal to zero except in those indices -2^i , $i = 0, 1, \dots, k-1$, where -2^i is taken modulo 2^k-1 , and whose GT coefficients satisfy nontrivial conjugacy relations.

3.3 Linearized Functions

If the field to which the function values of a LGSF belong (which is $GF(2^n)$ or a subfield of it) is same as the field to which the coefficients belong, then the conjugacy relations among the coefficients are trivial and we call such LGSFs simply as *linearized functions (LFs)* and the LGPs representing them as *linearized polynomials (LPs)*, to distinguish them from LFFs and LFPs respectively.

Definition 3.3.1: A LF is a LGSF representing a linear mapping from $GF(2^k)$ to $GF(2^n)$, and described by a LP, where a LP is an LGP whose coefficients satisfy trivial conjugacy relations and are independent of each other.

Note: A LF is a LGSF whose LGP representation satisfy trivial conjugacy relations among its coefficients. It can be described in terms of its Galois spectrum as follows:

A LF is a signal vector of length 2^k over $GF(2^n)$ whose Galois spectrum is identically equal to zero except in those indices -2^i , $i = 0, 1, \dots, k-1$, where -2^i is taken modulo 2^k-1 , and whose GT coefficients satisfy trivial conjugacy relations.

3.4 The Class of Linearized GSFs as an Ideal in Monoid Algebra

It may be shown that the class of LGSFs forms *an ideal in the cyclic monoid algebra of GSFs*. To show this result, we recall the transform domain description of GSFs discussed in Chapter 2. It is easy to see that pointwise addition of the GP coefficients representing two LGSFs gives another LGSF. In other words, LGSFs are closed under pointwise addition. Similarly, consider the pointwise multiplication of the GP coefficients of a LGSF with that of any GSF. This obviously results in the product being a LGSF, since pointwise multiplication forces all the remaining GP coefficients in the product, other than the linearized terms, to zero. Thus the class of LGSFs constitutes an ideal in the cyclic monoid algebra of GSFs.

Next we shall establish a one-to-one correspondence between LGSFs and linear transformations. But before doing this, we first define linear (n,k) transformations and take up some of the related counting problems. Our studies are limited to finite fields of characteristic 2.

5 Linear (n,k) Transformations

We define a linear (n,k) transformation as follows:

Definition 3.5.1: A linear (n,k) transformation is a linear transformation from a vector space $GF(2^k)$ to the vector space $GF(2^n)$, where k is not necessarily equal to n.

In general, the linear transformation may be one-to-one or many-to-one in nature. If $k = n$, and the mapping is one-to-one, the corresponding linear transformation represents a permutation of the vector space $GF(2^n)$. If $k < n$, and the mapping is one-to-one, the corresponding linear transformation represents a k dimensional subspace of the vector space $GF(2^n)$, which qualifies to be a linear (n,k) block code.

3.5.1 Number of Possible Linear (n,k) Transformations for a given pair of n and k

The number of possible linear (n,k) transformations (not necessarily one-to-one) for a given pair of n and k may be found as follows:

Every linear (n,k) transformation may be generated by a set of k vectors belonging to $GF(2^n)$, not necessarily linearly independent. Since each of the k vectors can assume any of the 2^n values, the number of possible linear (n,k) transformations for a given pair of n and k, is equal to $2^{n.k}$.

3.5.2 Number of Linear (n,k) Transformations Representing One-to-One Mappings (N_n)

If the set of k vectors $\in GF(2^n)$ which generates the linear (n,k) transformation are linearly independent, then the same would represent a one-to-one mapping, and hence a k-dimensional subspace of $GF(2^n)$. The number of such linear (n,k) transformations (N_n) representing one-to-one mappings is equal to the number of ways of choosing a k-dimensional subspace of an n-dimensional vector space over $GF(2)$, and the same is

obtained as follows [33]:

There are $2^n - 1$ ways of choosing the first basis element, since the 0 element cannot be chosen for a basis. There are $2^n - 2$ ways of choosing the second basis element, excluding the 0 element and the first basis element. Similarly, there are $2^n - 2^2$ ways of choosing the third basis element, since we exclude the 0 element and linear combinations (2^2 in number) of the first two. In the same way, the k^{th} basis element can be chosen in $2^n - 2^{k-1}$ ways.

Thus

$$N_n = (2^n - 1)(2^n - 2)(2^n - 2^2) \dots (2^n - 2^{k-1}). \quad (3.5.1)$$

3.5.3 Number of Distinct Linear (n,k) Transformations Representing One-to-One Mappings (N_{dist})

This number is calculated as follows [33]:

The number of ways of choosing a k -dimensional subspace of an n -dimensional vector space over $\text{GF}(2)$, is equal to N_n , as given in the previous section. Now, each such k -dimensional subspace containing 2^k n -tuples can be generated in N_k different ways, where N_k is given by

$$N_k = (2^k - 1)(2^k - 2)(2^k - 2^2) \dots (2^k - 2^{k-1}). \quad (3.5.2)$$

Thus the number of distinct k -dimensional subspaces, equal to the number of distinct linear (n,k) transformations representing one-to-one mappings, say N_{dist} , is given by

$$N_{\text{dist}} = N_n / N_k. \quad (3.5.3a)$$

A simplified expression for N_{dist} may be obtained as follows:

We may write N_k as

$$N_k = 2^{(k-1)k/2} \prod_{i=1}^k (2^i - 1).$$

Similarly, N_n may be written as

$$N_n = 2^{(k-1)k/2} \prod_{i=n-k+1}^n (2^i - 1).$$

$$\text{hus } N_{\text{dist}} = N_n / N_k = \left[\prod_{i=n-k+1}^n (2^i - 1) \right] / \left[\prod_{i=1}^k (2^i - 1) \right]. \quad (3.5.3b)$$

.6 Correspondence Between Linearized GSFs and Linear (n,k) Transformations

We show in the following theorem that there exists a one-to-one correspondence between LGSFs representing linear mappings from $GF(2^k)$ to $GF(2^n)$, and linear (n,k) transformations, for given pair of n and k.

Theorem 3.6.1: A linear (n,k) transformation over $GF(2)$, can be represented by a LGSF of the form (3.1.3) given by

$$f(x) = \sum_{i=0}^{k-1} a_{-2^i} x^{2^i},$$

where $f(x) \in GF(2^n)$, $x \in GF(2^k)$, $a_{-2^i} \in GF(2^L)$, $i = 0, 1, \dots, k-1$, $L = \text{L.C.M. of } n \text{ and } k$, and -2^i is taken modulo $2^k - 1$.

Proof: A linear (n,k) transformation is a mapping from $GF(2^k)$ to $GF(2^n)$. Hence it can be represented by a GSF described by a LGP with coefficients from $GF(2^L)$, L being the L.C.M. of n and k.

To prove that the above is a LGSF, let us consider two k-tuples x_1 and $x_2 \in GF(2^k)$. Let the corresponding n-tuples generated by the transformation which belong to $GF(2^n)$ be $f(x_1)$ and $f(x_2)$ respectively. Now, since the transformation is linear, the n-tuple vector generated corresponding to the sum of the above two k-tuple vectors i.e., $x_1 + x_2$, should be equal to the sum of the corresponding n-tuple vectors generated, i.e., $f(x_1) + f(x_2)$. In other words, $f(x_1 + x_2)$ should be equal to $f(x_1) + f(x_2)$. According to the

discussions in Section 3.1.1, this is possible only if $f(x)$ is a LGSF.

Q.E.D.

3.7 Relating the Coefficients of a Linearized GP to the Vectors Generating the Corresponding Linear (n,k) Transformation

In the next theorem, we derive the relations between the LGP coefficients of a LGSF and the vectors generating the corresponding linear (n,k) transformation:

Theorem 3.7.1: The coefficients of the LGP representing a linear (n,k) transformation are related to the vectors which generate the transformation, by the relation $\underline{A} = \underline{V}^{-1} \underline{f}$, where \underline{V}^{-1} is the inverse of a Vander monde matrix \underline{V} , of size $k \times k$, of the form

$$\underline{V} = \begin{bmatrix} 1 & 1 & 1 & . & . & 1 \\ \alpha & \alpha^2 & \alpha^{2^2} & . & . & \alpha^{2^{k-1}} \\ (\alpha)^2 & (\alpha^2)^2 & (\alpha^{2^2})^2 & . & . & (\alpha^{2^{k-1}})^2 \\ (\alpha)^3 & (\alpha^2)^3 & (\alpha^{2^2})^3 & . & . & (\alpha^{2^{k-1}})^3 \\ . & . & . & . & . & . \\ (\alpha)^{k-1} & (\alpha^2)^{k-1} & (\alpha^{2^2})^{k-1} & . & . & (\alpha^{2^{k-1}})^{k-1} \end{bmatrix} \quad (3.7.1)$$

where \underline{A} is the coefficient vector $\in GF(2^L)$, $L = \text{L.C.M. of } n \text{ and } k$, of length k , \underline{f} a k -length vector whose components belonging to $GF(2^n)$ are the vectors which generate the linear transformation, and α is a primitive element of $GF(2^k)$.

Proof: The LGSF representing the linear (n,k) transformation is described by (3.1.3).

Substituting $x = \alpha^i$, $i = 0, 1, \dots, k-1$, in (3.1.3) we get the following relation:

$$\begin{bmatrix} f(\alpha^0) \\ f(\alpha) \\ f(\alpha^2) \\ . \\ . \\ f(\alpha^{k-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & . & . & 1 \\ \alpha & \alpha^2 & \alpha^{2^2} & . & . & \alpha^{2^{k-1}} \\ (\alpha)^2 & (\alpha^2)^2 & (\alpha^{2^2})^2 & . & . & (\alpha^{2^{k-1}})^2 \\ (\alpha)^3 & (\alpha^2)^3 & (\alpha^{2^2})^3 & . & . & (\alpha^{2^{k-1}})^3 \\ . & . & . & . & . & . \\ (\alpha)^{k-1} & (\alpha^2)^{k-1} & (\alpha^{2^2})^{k-1} & . & . & (\alpha^{2^{k-1}})^{k-1} \end{bmatrix} \begin{bmatrix} a_{-2^0} \\ a_{-2^1} \\ . \\ . \\ . \\ a_{-2^{k-1}} \end{bmatrix}$$

$$\text{or } \underline{f} = \underline{V} \underline{A} \quad (3.7.2)$$

It may be readily seen that the $k \times k$ matrix on the right hand side, \underline{V} , has the structure of a *Vander monde matrix*, with the distinct elements being α^{2^i} , $i = 0, 1, \dots, k-1$. Since \underline{V} is always known to be invertible, we can write $\underline{A} = \underline{V}^{-1} \underline{f}$.

Q.E.D.

In the following two corollaries, we study the nature of the matrix \underline{V}^{-1} when standard basis (SB) and normal basis (NB) are respectively employed for representing the elements of $GF(2^k)$. In the former case, we denote \underline{V}^{-1} as \underline{V}_s^{-1} and in the latter as \underline{V}_n^{-1} .

Corollary 3.7.1: The inverse of the Vander monde matrix in (3.7.1), ie., \underline{V}_s^{-1} , is of the form

$$\underline{V}_s^{-1} = \begin{bmatrix} b_{00} & b_{01} & b_{02} & \dots & b_{0 \ k-1} \\ (b_{00})^2 & (b_{01})^2 & (b_{02})^2 & \dots & (b_{0 \ k-1})^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (b_{00})^{2^{k-1}} & (b_{01})^{2^{k-1}} & (b_{02})^{2^{k-1}} & \dots & (b_{0 \ k-1})^{2^{k-1}} \end{bmatrix} \quad (3.7.3)$$

if SB is used for representing the elements of $GF(2^k)$,

$$\text{where } b_{0t} = \sum_{j=0}^{\zeta} m_{jt} \alpha^j \in GF(2^k), \quad t = 0, 1, \dots, k-1; \quad (3.7.4)$$

$\zeta = 2^k - 2$, α is a primitive element of $GF(2^k)$ in SB, α^j is represented in SB as

$$\alpha^j = m_{j \ k-1} \alpha^{k-1} + \dots + m_{j2} \alpha^2 + m_{j1} \alpha^1 + m_{j0} \alpha^0, \quad (3.7.5)$$

α^t , $t = 0, 1, \dots, k-1$, being the SB vectors used for representing $GF(2^k)$, and $m_{jt} \in \{0,1\}$, is the coefficient of α^t .

Proof: The coefficients a_t are related to the function values $f(\cdot)$ by the relation

$$a_t = \sum_{x \in GF(2^k)} x^t f(x), \quad t = 0, 1, 2, \dots, 2^k - 2.$$

This can also be expressed in terms of α , the primitive element of $GF(2^k)$, as

$$a_t = \sum_{j=0}^{\zeta} (\alpha^t)^j f(\alpha^j), \quad t = 0, 1, 2, \dots, 2^k - 2.$$

We are interested only in those $t = 2^k - 1 - 2^i = -2^i \pmod{2^k - 1}$, $i = 0, 1, \dots, k-1$, since we are considering LGSFs.

Thus the coefficients of the LGP representing the linear (n,k) transformation, are given by

$$a_{-2^i} = \sum_{j=0}^{\zeta} (\alpha^{-2^i})^j f(\alpha^j), \quad i = 0, 1, 2, \dots, k-1. \quad (3.7.6)$$

Since $f(\alpha^j)$, $j = 0, 1, \dots, k-1$, are the vectors which generate the transformation, $f(\alpha^j)$, $j = k, k+1, \dots, 2^k-2$, can be expressed as a linear combination of $f(\alpha^j)$, $j = 0, 1, \dots, k-1$. The linear combination depends on the modulo polynomial chosen for generating $GF(2^k)$. Thus

$$f(\alpha^j) = \sum_{t=0}^{k-1} m_{jt} f(\alpha^t) \quad (3.7.7)$$

where $m_{jt} \in \{0,1\}$, is the coefficient of x^t , in the polynomial representation of α^j , in x of degree $\leq k-1$, in SB.

Therefore,

$$\begin{aligned} a_{-2^i} &= \sum_{j=0}^{\zeta} (\alpha^{-2^i})^j \left(\sum_{t=0}^{k-1} m_{jt} f(\alpha^t) \right) \\ &= \sum_{t=0}^{k-1} \sum_{j=0}^{\zeta} m_{jt} (\alpha^{-j})^{2^i} f(\alpha^t) \\ &= \sum_{t=0}^{k-1} \sum_{j=0}^{\zeta} (m_{jt} \alpha^{-j})^{2^i} f(\alpha^t), i = 0, 1, \dots, k-1. \end{aligned} \quad (3.7.8)$$

Thus \underline{V}_s^{-1} will consist of elements of the form

$$b_{it} = \sum_{j=0}^{\zeta} (m_{jt} \alpha^{-j})^{2^i} \in GF(2^k), i, t = 0, 1, \dots, k-1. \quad \text{Q.E.D.}$$

We have thus seen in Corollary 3.7.1 that it is necessary only to calculate the first row of the matrix \underline{V}_s^{-1} . However, the matrix \underline{V}^{-1} has an even simpler structure, if NB is used for representing the elements of $GF(2^k)$, rather than SB, as shown in the next corollary:

Corollary 3.7.2: The inverse of the Vander monde matrix, ie., \underline{V}_n^{-1} , is of the form

$$\underline{V}_n^{-1} = \begin{bmatrix} b_0 & b_0^2 & b_0^{2^2} & \cdot & \cdot & b_0^{2^{k-1}} \\ b_0^2 & b_0^{2^2} & b_0^{2^3} & \cdot & \cdot & b_0 \\ b_0^{2^2} & b_0^{2^3} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_0^{2^{k-1}} & b_0 & b_0^2 & \cdot & \cdot & b_0^{2^{k-2}} \end{bmatrix} \quad (3.7.9)$$

,if NB is used for representing the elements of $GF(2^k)$,

$$\text{where} \quad b_0 = \sum_{j=0}^{\xi} m_{j0} \delta^j \in GF(2^k); \quad (3.7.10)$$

$\xi = 2^k - 2$, δ is a primitive element of $GF(2^k)$ in NB, δ^j is represented in NB as

$$\delta^j = m_{j \ k-1} \Theta^{2^{k-1}} + \dots + m_{j2} \Theta^{2^2} + m_{j1} \Theta^2 + m_{j0} \Theta, \quad (3.7.11)$$

Θ^{2^t} , $t = 0, 1, \dots, k-1$, being the NB vectors used for representing $GF(2^k)$, $m_{jt} \in \{0, 1\}$, is the coefficient of Θ .

Proof: To prove this corollary, we need prove that the elements in the first row of \underline{V}_s^{-1} , given in Corollary 3.7.1, have the additional relation that $b_{0t} = (b_{00})^{2^t}$, $t = 1, 2, \dots, k-1$.

Substituting for b_{0t} and b_{00} respectively in the above equation, we get

$$\begin{aligned} \sum_{j=0}^{\xi} m_{jt} \delta^j &= \left(\sum_{j=0}^{\xi} m_{j0} \delta^j \right)^{2^t} \\ &= \sum_{j=0}^{\xi} (m_{j0} \delta^{j \cdot 2^t}), \quad t = 1, \dots, k-1, ; \xi = 2^k - 2 \end{aligned}$$

(the exponent $j \cdot 2^t$ taken modulo $2^k - 1$).

Expanding the summation terms on both sides,

$$m_{0t} \delta^0 + m_{1t} \delta^1 + m_{2t} \delta^2 + m_{3t} \delta^3 + \dots + m_{2^{k-2}t} \delta^{(2^{k-2})}$$

$$= m_{00} \delta^{-0.2^t} + m_{10} \delta^{-1.2^t} + m_{20} \delta^{-2.2^t} + m_{30} \delta^{-3.2^t} + \dots + m_{2^{k-2}0} \delta^{-(2^{k-2}).2^t} \quad (3.7.12)$$

from which it is evident that

$$m_{j.2^t t} = m_{j0}, \quad (3.7.13)$$

should be true in order that the required relation be satisfied. Hence we prove the relation given by (3.7.13) as follows:

$$\begin{aligned} \text{We have } \delta^j &= m_{j k-1} \Theta^{2^{k-1}} + \dots + m_{j2} \Theta^{2^2} + m_{j1} \Theta^2 + m_{j0} \Theta, \\ \text{and } \delta^{j.2^t} &= m_{2^t j k-1} \Theta^{2^{k-1}} + \dots + m_{2^t j i} \Theta^{2^i} + \dots + m_{2^t j 1} \Theta^2 + m_{2^t j 0} \Theta, \end{aligned} \quad (3.7.14a)$$

Now $\delta^{j.2^t}$ can also be expressed in terms of the coefficients of δ^j , with the latter cyclically shifted to the left by t places, since each squaring in NB cyclically shifts the coefficients left by one place.

Thus $\delta^{j.2^t}$ can also be written as

$$\delta^{j.2^t} = m_{j k-1-t} \Theta^{2^{k-1}} + \dots + m_{j i-t} \Theta^{2^i} + \dots + m_{j k-1+t} \Theta^2 + m_{j k-t} \Theta, \quad (3.7.14b)$$

where the second subscript of m is taken modulo k .

Comparing (3.7.14a) and (3.7.14b), we can write $m_{2^t j i} = m_{j i-t}$.

Putting $i = t$, we get (3.7.13).

Q.E.D.

3.8 Conjugacy Relations in Linearized GPs

In the following theorems, we discuss the conjugacy relations in LGPs. As in the general case, here also we study these relations under two broad classifications, namely, those LGSFs whose $k \nmid n$, and those whose $k | n$. The relations are same as in the general case, except that now, in general, only k coefficients will be nonzero, and hence the expressions are simplified.

(a) $k \nmid n$

Theorem 3.8.1: If $k \nmid n$, then a linear (n,k) transformation can be represented, in general, by a *linearized Frobenius polynomial* (LFP), as

$$f(x) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i x^{2^i}), \quad (3.8.1)$$

where $\gamma_i \in \text{GF}(2^L)$, $\text{frs}(\theta) = \theta + \theta^Q + \theta^{Q^2} + \dots + \theta^{Q^{t-1}}$, $\theta^{Q^t} = \theta$, $Q = 2^n$, $L = \text{L.C.M}$ of n and k , $g = \text{G.C.D.}$ of n and k , and $t = L/n$,

(1) if the vectors in the linear transformation belong to $\text{GF}(2^n)$ and not to any of its subfields, and

(2) n is replaced by n_1 , if the the vectors in the linear transformation belong to $\text{GF}(2^{n_1})$ as well as to a subfield of it, namely, $\text{GF}(2^{n_1})$.

Proof: If $k \nmid n$, then $L \neq n$. Hence the coefficients belong to an extension of $\text{GF}(2^n)$, the extension order being, say, $t = L/n$. The conjugacy constraints are nontrivial in this case. The k terms in the LGP can be grouped into Frobenius cycles, each cycle containing t terms. The number of such Frobenius cycles will be thus equal to $\frac{k}{t} = \frac{k}{L/n} = nk/L = g = \text{G.C.D.}$ of n and k , since any n and k satisfy the relation

$$n.k = \text{G.C.D}(n,k) \cdot \text{L.C.M}(n,k). \quad (3.8.2)$$

If the code vectors $\in \text{GF}(2^{n_1})$, a subfield of $\text{GF}(2^n)$, then (3.8.1) is valid with n replaced by n_1 , since now the mapping is from $\text{GF}(2^k)$ to $\text{GF}(2^{n_1})$.
Q.E.D.

Examples: Let us consider some examples of LFP representations of linear (n,k) transformations where $k \nmid n$.

Example 3.8.1: First let us consider the case where the G.C.D. of n and $k = 1$, ie., n and k are relatively prime and the vectors in the linear transformation $\in \text{GF}(2^n)$. So let $n = 5$,

$k = 3$. $L = \text{L.C.M. of } n \text{ and } k = \text{L.C.M. of } 5 \text{ and } 3 = 15$.

In this example, a one-to-one linear transformation, i.e., a linear code, is considered. We denote the primitive elements of $\text{GF}(2^3)$, $\text{GF}(2^5)$ and $\text{GF}(2^{15})$ as α , β and γ respectively. We choose the minimal polynomial for generating $\text{GF}(2^{15})$ (which is the minimal polynomial of γ) as $x^{15} + x + 1$. Then the minimal polynomial for generating the subfields $\text{GF}(2^3)$ and $\text{GF}(2^5)$ are respectively the minimal polynomials of γ^{ν_1} and γ^{ν_2} where $\nu_1 = \frac{2^{15}-1}{2^3-1} = 4681$ and $\nu_2 = \frac{2^{15}-1}{2^5-1} = 1057$ which are $x^3 + x + 1$ and $x^5 + x^3 + x^2 + x + 1$.

Let the Generator matrix \underline{G} for the code be $= \begin{bmatrix} 01110 \\ 00101 \\ 11110 \end{bmatrix}$.

Using the SB table for $\text{GF}(2^3)$ generated by $x^3 + x + 1$ (not listed), we calculate

$$b_{00} = \sum_{j=0}^6 m_{j0} \alpha^{-j} = \alpha^{-0} + \alpha^{-3} + \alpha^{-5} + \alpha^{-6} = 1 = \gamma^0.$$

$$b_{01} = \sum_{j=0}^6 m_{j1} \alpha^{-j} = \alpha^{-1} + \alpha^{-3} + \alpha^{-4} + \alpha^{-5} = \alpha^2 = \gamma^{9362}.$$

$$b_{02} = \sum_{j=0}^6 m_{j2} \alpha^{-j} = \alpha^{-2} + \alpha^{-4} + \alpha^{-5} + \alpha^{-6} = \alpha = \gamma^{4681}.$$

$$y_0 = 11110 = \beta^6 = \gamma^{6342}, y_1 = 00101 = \beta^{24} = \gamma^{25368}, y_2 = 01110 = \beta^{28} = \gamma^{29596}.$$

The coefficients of the LGP (which is an LFP) representing this code corresponding to the given basis, are given by

$$\begin{bmatrix} a_6 \\ a_5 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & \gamma^{9362} & \gamma^{4681} \\ 1 & \gamma^{18724} & \gamma^{9362} \\ 1 & \gamma^{4681} & \gamma^{18724} \end{bmatrix} \begin{bmatrix} \gamma^{6342} \\ \gamma^{25368} \\ \gamma^{29596} \end{bmatrix} = \begin{bmatrix} \gamma^{6357} \\ \gamma^{21702} \\ \gamma^{6822} \end{bmatrix}$$

$$\text{Thus } f_8(x) = \gamma^{6822} x^4 + \gamma^{21702} x^2 + \gamma^{6357} x,$$

which can be expressed as a single term LFP as

$$f_8(x) = \text{frs}(\gamma^{6357} x),$$

since the coefficients which belong to $\text{GF}(2^{15})$, satisfy conjugacy relations given by

$$(a_6)^{32} = a_3; (a_3)^{32} = a_5; (a_5)^{32} = a_6 \text{ (modulo 32767)}.$$

Example 3.8.2: Let us consider a linear (n,k) code where the G.C.D. of n and $k = 2$. So let $n = 6, k = 4; L = \text{L.C.M. of } 6 \text{ and } 4 = 12$.

We denote the primitive elements of $\text{GF}(2^4)$, $\text{GF}(2^6)$ and $\text{GF}(2^{12})$ as α , β and γ respectively. We choose the minimal polynomial for generating $\text{GF}(2^{12})$ (minimal polynomial of γ) as $x^{12} + x^6 + x^4 + x + 1$. Then the minimal polynomial for generating the subfields $\text{GF}(2^4)$ and $\text{GF}(2^6)$ are the minimal polynomials of γ^{ν_1} and γ^{ν_2} where $\nu_1 = \frac{2^{12}-1}{2^4-1} = 273$ and $\nu_2 = \frac{2^{12}-1}{2^6-1} = 65$ which are respectively $x^4 + x + 1$ and $x^6 + x^5 + 1$.

Let the Generator matrix \underline{G} for the code be $= \begin{bmatrix} 111111 \\ 001001 \\ 001110 \\ 100011 \end{bmatrix}$.

Using the SB table for $\text{GF}(2^4)$ (given in Appendix C.3), we calculate

$$b_{00} = \sum_{j=0}^{14} m_{j0} \alpha^{-j} = \alpha^{-0} + \alpha^{-4} + \alpha^{-7} + \alpha^{-8} + \alpha^{-10} + \alpha^{-12} + \alpha^{-13} + \alpha^{-14} = \alpha^{14} = \gamma^{3822}.$$

$$b_{01} = \sum_{j=0}^{14} m_{j1} \alpha^{-j} = \alpha^{-1} + \alpha^{-4} + \alpha^{-5} + \alpha^{-7} + \alpha^{-9} + \alpha^{-10} + \alpha^{-11} + \alpha^{-12} = \alpha^2 = \gamma^{546}.$$

$$b_{02} = \sum_{j=0}^{14} m_{j2} \alpha^{-j} = \alpha^{-2} + \alpha^{-5} + \alpha^{-6} + \alpha^{-8} + \alpha^{-10} + \alpha^{-11} + \alpha^{-12} + \alpha^{-13} = \alpha = \gamma^{273}.$$

$$b_{03} = \sum_{j=0}^{14} m_{j3} \alpha^{-j} = \alpha^{-3} + \alpha^{-6} + \alpha^{-7} + \alpha^{-9} + \alpha^{-11} + \alpha^{-12} + \alpha^{-13} + \alpha^{-14} = \alpha^0 = 1.$$

$$y_0 = 1000111 = \beta^7 = \gamma^{455}, y_1 = 001110 = \beta^{40} = \gamma^{2600}, y_2 = 001001 = \beta^{34} = \gamma^{2210}, y_3 = 111111 = \beta^{10} = \gamma^{650}.$$

The coefficients can be computed as

$$\begin{bmatrix} a_{14} \\ a_{13} \\ a_{11} \\ a_7 \end{bmatrix} = \begin{bmatrix} \gamma^{3822} & \gamma^{546} & \gamma^{273} & 1 \\ \gamma^{3549} & \gamma^{1092} & \gamma^{546} & 1 \\ \gamma^{3003} & \gamma^{2184} & \gamma^{1092} & 1 \\ \gamma^{1911} & \gamma^{273} & \gamma^{2184} & 1 \end{bmatrix} \begin{bmatrix} \gamma^{455} \\ \gamma^{2600} \\ \gamma^{2210} \\ \gamma^{650} \end{bmatrix} = \begin{bmatrix} \gamma^{470} \\ \gamma^{458} \\ \gamma^{1415} \\ \gamma^{647} \end{bmatrix}$$

which can be expressed as an LFP containing two Frobenius terms as

$$f_s(x) = \text{frs}(\gamma^{470} x) + \text{frs}(\gamma^{458} x^2),$$

where $\text{frs}(\Theta) = \Theta + \Theta^{64}$.

Example 3.8.3: We note that the number of Frobenius terms in the LFP representation can be less than the G.C.D. of n and k . But the maximum number of Frobenius terms is equal to the G.C.D. of n and k . For example, linear $(6,4)$ codes with their LFP representations having single Frobenius terms can exist, even though the maximum number of possible terms in this case is 2.

We illustrate this case in this example. Let the primitive elements of $\text{GF}(2^4)$, $\text{GF}(2^6)$ and $\text{GF}(2^{12})$ and the minimal polynomials for generating $\text{GF}(2^{12})$, $\text{GF}(2^4)$ and $\text{GF}(2^6)$ be the same as in Example 3.8.2.

Let the Generator matrix \underline{G} be
$$\begin{bmatrix} 001010 \\ 011111 \\ 110001 \\ 100001 \end{bmatrix}.$$

As in the previous example, $b_{00} = \alpha^{14} = \gamma^{3822}$; $b_{01} = \alpha^2 = \gamma^{546}$; $b_{02} = \alpha = \gamma^{273}$; $b_{03} = \alpha^0 = 1$.

$$y_0 = 100001 = \beta^6 = \gamma^{390}, y_1 = 110001 = \beta^{57} = \gamma^{3705}, y_2 = 011111 = \beta^{11} = \gamma^{715}, \\ y_3 = 001010 = \beta^{54} = \gamma^{3510}.$$

The coefficients can be computed as

$$\begin{bmatrix} a_{14} \\ a_{13} \\ a_{11} \\ a_7 \end{bmatrix} = \begin{bmatrix} \gamma^{3822} & \gamma^{546} & \gamma^{273} & 1 \\ \gamma^{3549} & \gamma^{1092} & \gamma^{546} & 1 \\ \gamma^{3003} & \gamma^{2184} & \gamma^{1092} & 1 \\ \gamma^{1911} & \gamma^{273} & \gamma^{2184} & 1 \end{bmatrix} \begin{bmatrix} \gamma^{390} \\ \gamma^{3705} \\ \gamma^{715} \\ \gamma^{3510} \end{bmatrix} = \begin{bmatrix} \gamma^{587} \\ 0 \\ \gamma^{713} \\ 0 \end{bmatrix}$$

which can be expressed as a single term LFP as

$$fs(x) = frs(\gamma^{587} x),$$

where $frs(\Theta) = \Theta + \Theta^{64}$.

Example 3.8.4: We consider an example of a many-to-one linear (n,k) transformation, with $n = 6$ and $k = 4$. Let the primitive elements of $GF(2^4)$, $GF(2^6)$ and $GF(2^{12})$ and the minimal polynomials for generating $GF(2^{12})$, $GF(2^4)$ and $GF(2^6)$ be the same as in Example 3.8.2.

Let the set of vectors which generate the transformation be given by

$$\begin{bmatrix} 001001 \\ 001110 \\ 000111 \\ 100011 \end{bmatrix}.$$

From the previous example, $b_{00} = \gamma^{3822}$; $b_{01} = \gamma^{546}$; $b_{02} = \gamma^{273}$; $b_{03} = 1$.

Now, $y_0 = 100011 = \beta^7 = \gamma^{455}$, $y_1 = 000111 = \beta^{39} = \gamma^{2535}$, $y_2 = 001110 = \beta^{40} = \gamma^{2600}$, $y_3 = 001001 = \beta^{34} = \gamma^{2210}$.

The coefficients can be computed as

$$\begin{bmatrix} a_{14} \\ a_{13} \\ a_{11} \\ a_7 \end{bmatrix} = \begin{bmatrix} \gamma^{3822} & \gamma^{546} & \gamma^{273} & 1 \\ \gamma^{3549} & \gamma^{1092} & \gamma^{546} & 1 \\ \gamma^{3003} & \gamma^{2184} & \gamma^{1092} & 1 \\ \gamma^{1911} & \gamma^{273} & \gamma^{2184} & 1 \end{bmatrix} \begin{bmatrix} \gamma^{455} \\ \gamma^{2535} \\ \gamma^{2600} \\ \gamma^{2210} \end{bmatrix} = \begin{bmatrix} \gamma^{3836} \\ \gamma^{2770} \\ \gamma^{3899} \\ \gamma^{1195} \end{bmatrix}$$

which can be expressed as a LFP containing two Frobenius terms as

$$\text{fs}(x) = \text{fs}(\gamma^{3836} x) + \text{fs}(\gamma^{2770} x^2).$$

where $\text{fs}(\Theta) = \Theta + \Theta^{64}$.

Example 3.8.5: Finally, we give a many-to-one linear (6,4) transformation, in which the vectors in the linear transformation belong to a subfield of $\text{GF}(2^6)$, say $\text{GF}(2^3)$. Let the primitive elements of $\text{GF}(2^4)$, $\text{GF}(2^6)$ and $\text{GF}(2^{12})$ and the minimal polynomials for generating $\text{GF}(2^{12})$, $\text{GF}(2^4)$ and $\text{GF}(2^6)$ be the same as in Example 3.8.2.

Let the set of vectors which generate the transformation be given by

$$\begin{bmatrix} 101110 \\ 100101 \\ 001011 \\ 101111 \end{bmatrix}.$$

b_{ij} 's had been obtained as, $b_{00} = \gamma^{3822}$; $b_{01} = \gamma^{546}$; $b_{02} = \gamma^{273}$; $b_{03} = 1$.

Now, $y_0 = 101111 = \beta^9 = \gamma^{585}$, $y_1 = 001011 = \beta^{18} = \gamma^{1170}$, $y_2 = 100101 = \beta^{45} = \gamma^{2925}$, $y_3 = 101110 = \beta^{27} = \gamma^{1755}$. It may be noted that the y_i 's also $\in \text{GF}(2^3)$.

The coefficients can be computed as

$$\begin{bmatrix} a_{14} \\ a_{13} \\ a_{11} \\ a_7 \end{bmatrix} = \begin{bmatrix} \gamma^{3822} & \gamma^{546} & \gamma^{273} & 1 \\ \gamma^{3549} & \gamma^{1092} & \gamma^{546} & 1 \\ \gamma^{3003} & \gamma^{2184} & \gamma^{1092} & 1 \\ \gamma^{1911} & \gamma^{273} & \gamma^{2184} & 1 \end{bmatrix} \begin{bmatrix} \gamma^{585} \\ \gamma^{1170} \\ \gamma^{2925} \\ \gamma^{1755} \end{bmatrix} = \begin{bmatrix} \gamma^{3843} \\ \gamma^{2016} \\ \gamma^{252} \\ \gamma^{2079} \end{bmatrix}$$

which can be expressed as a single term LFP as

$$f_s(x) = \text{frs}(\gamma^{3843} x),$$

where $\text{frs}(\Theta) = \Theta + \Theta^8$.

(b) $k|n$

Theorem 3.8.2: If $k|n$, then the conjugacy relations among the k coefficients of the LGP representing the linear (n,k) transformation are trivial, and the function represented as a linearized function (LF) if

- (1) the vectors in the linear transformation belong to $\text{GF}(2^n)$ and not to any subfield of it, and if
- (2) the vectors in the linear transformation belong to $\text{GF}(2^n)$ as well as to a subfield of it, namely, $\text{GF}(2^{n_1})$, where $k|n_1$.

The coefficients belong to $\text{GF}(2^n)$ and $\text{GF}(2^{n_1})$ respectively in (1) and (2).

Conjugacy relations exist and the linear transformation can be represented by a LFP of the form (3.8.1), if the vectors in the linear transformation belong to $\text{GF}(2^n)$ as well as to a subfield of it, namely, $\text{GF}(2^{n_1})$, where $k \nmid n_1$.

Proof: If $k|n$, and the vectors in the transformation belong to $\text{GF}(2^n)$, but not to any subfield of it, then $L = n$. Hence the coefficients belong to $\text{GF}(2^n)$. On the other hand, if the vectors belong to $\text{GF}(2^{n_1})$, a subfield of $\text{GF}(2^n)$ and $k|n_1$, then $L = n_1$. In both cases, since the coefficients and the function values belong to the same field, any coefficient raised

to the Q^{th} power ($Q = 2^n$ or 2^{n_1} as the case may be) results in the same coefficient. Thus the conjugacy relations are trivial and all the coefficients are independent of each other, thus the resulting LGSF is a LF.

However, if $k \nmid n_1$, then $L \neq n_1$, and therefore $\text{GF}(2^{n_1})$ is a proper subfield of $\text{GF}(2^L)$. Thus the function may be represented by (3.8.1) with n replaced by n_1 .

Q.E.D.

Examples: Let us consider some examples of LGSF representations of linear (n,k) transformations where $k|n$.

Example 3.8.6: In this example, we consider the LGSF representation of a linear (n,k) transformation which represents a one-to-one mapping, (ie., a linear (n,k) code) where $k|n$. Let $n = 4$, $k = 2$.

$L = \text{L.C.M. of } n \text{ and } k = \text{L.C.M. of } 4 \text{ and } 2 = 4$.

We denote the primitive elements of $\text{GF}(2^2)$ and $\text{GF}(2^4)$ as α and β respectively. We choose the minimal polynomial for generating $\text{GF}(2^4)$ (which is the minimal polynomial of θ) as $x^4 + x + 1$. Then the minimal polynomial for generating the subfield $\text{GF}(2^2)$ is the minimal polynomial of β^ν where $\nu = \frac{2^4-1}{2^2-1} = 5$ which is $x^2 + x + 1$.

We choose the Generator matrix \underline{G} for the code as $\begin{bmatrix} y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} 1100 \\ 0011 \end{bmatrix}$.

Using the SB table for $\text{GF}(2^2)$ (Appendix C.1), we calculate

$$b_{00} = \sum_{j=0}^2 m_{j0} \alpha^{-j} = 1 \cdot \alpha^{-0} + 0 \cdot \alpha^{-1} + 1 \cdot \alpha^{-2} = \alpha^0 + \alpha = \alpha^2 = \beta^{10}.$$

$$b_{01} = \sum_{j=0}^2 m_{j1} \alpha^{-j} = 0 \cdot \alpha^{-0} + 1 \cdot \alpha^{-1} + 1 \cdot \alpha^{-2} = \alpha^2 + \alpha = \alpha^0 = 1.$$

$$y_0 = 0011 = \beta^4, y_1 = 1100 = \beta^6.$$

The coefficients of the LGP representing this code corresponding to the given basis, are given by

$$\begin{bmatrix} a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} \beta^{10} & 1 \\ \beta^5 & 1 \end{bmatrix} \begin{bmatrix} \beta^4 \\ \beta^6 \end{bmatrix} = \begin{bmatrix} \beta^8 \\ \beta^5 \end{bmatrix}$$

We see that the coefficients are independent of each other and the code can be represented by a LP as

$$f_s(x) = \beta^5 x^2 + \beta^8 x.$$

Example 3.8.7: In this example, we consider the LGSF representation of a linear (n,k) transformation which represents a many-to-one mapping, where $k|n$. Let $n = 6$, $k = 3$.

$L = \text{L.C.M. of } 6 \text{ and } 3 = 6$. Let the primitive elements of $\text{GF}(2^3)$ and $\text{GF}(2^6)$ be α and β respectively. Let the minimal polynomial for generating $\text{GF}(2^6)$ be chosen as $x^6 + x + 1$. Then the minimal polynomial for generating the subfield $\text{GF}(2^3)$ is the minimal polynomial of β^ν where $\nu = \frac{2^6-1}{2^3-1} = 9$ which is $x^3 + x^2 + 1$.

Let the set of vectors which generate this linear transformation be given by

$$\begin{bmatrix} y_2 \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} 110101 \\ 101100 \\ 011001 \end{bmatrix}.$$

Using the SB table for $\text{GF}(2^3)$ (Appendix C.2), we obtain

$$\begin{aligned} b_{00} &= \sum_{j=0}^6 m_{j0} \alpha^{-j} = 1.\alpha^{-0} + 0.\alpha^{-1} + 0.\alpha^{-2} + 1.\alpha^{-3} + 1.\alpha^{-4} + 1.\alpha^{-5} + 0.\alpha^{-6} \\ &= \alpha^0 + \alpha^4 + \alpha^3 + \alpha^2 = \alpha^4 = \beta^{36}. \end{aligned}$$

$$\begin{aligned} b_{01} &= \sum_{j=0}^6 m_{j1} \alpha^{-j} = 0.\alpha^{-0} + 1.\alpha^{-1} + 0.\alpha^{-2} + 0.\alpha^{-3} + 1.\alpha^{-4} + 1.\alpha^{-5} + 1.\alpha^{-6} \\ &= \alpha^6 + \alpha^3 + \alpha^2 + \alpha = \alpha^3 = \beta^{27}. \end{aligned}$$

$$\begin{aligned} b_{02} &= \sum_{j=0}^6 m_{j2} \alpha^{-j} = 0.\alpha^{-0} + 0.\alpha^{-1} + 1.\alpha^{-2} + 1.\alpha^{-3} + 1.\alpha^{-4} + 0.\alpha^{-5} + 1.\alpha^{-6} \\ &= \alpha^5 + \alpha^4 + \alpha^3 + \alpha = \alpha^5 = \beta^{45}. \end{aligned}$$

$$y_0 = 011001 = \beta^{45}, y_1 = 101100 = \beta^{37}, y_2 = 110101 = \beta^{22}.$$

The coefficients of the LGP representing this linear transformation corresponding to the given set of vectors, are given by

$$\begin{bmatrix} a_6 \\ a_5 \\ a_3 \end{bmatrix} = \begin{bmatrix} \beta^{36} & \beta^{27} & \beta^{45} \\ \beta^9 & \beta^{54} & \beta^{27} \\ \beta^{18} & \beta^{45} & \beta^{54} \end{bmatrix} \begin{bmatrix} \beta^{45} \\ \beta^{37} \\ \beta^{22} \end{bmatrix} = \begin{bmatrix} \beta^{41} \\ \beta^{24} \\ \beta^{52} \end{bmatrix}$$

We see that the coefficients are independent of each other and the code can be represented by a LP as

$$f_s(x) = \beta^{52} x^4 + \beta^{24} x^2 + \beta^{41} x.$$

Example 3.8.8: Let us take an example of a linear (n,k) code in which all the code vectors $\in GF(2^{n_1})$ where $k|n_1$. Let $n = 12$, $k = 3$ and $n_1 = 6$. Let α , β and γ be primitive elements of $GF(2^3)$, $GF(2^6)$ and $GF(2^{12})$ respectively.

Minimal polynomial for generating $GF(2^3)$: $x^3 + x^2 + 1$.

Minimal polynomial for generating $GF(2^{12})$: $x^{12} + x^{11} + x^8 + x^6 + 1$.

b_{ij} 's have been obtained as

$$b_{00} = \alpha^4 = \gamma^{2340}, b_{01} = \alpha^3 = \gamma^{1755}, b_{02} = \alpha^5 = \gamma^{2925}.$$

$$y_0 = 100011010111 = \gamma^{715} = \beta^{11}, y_1 = 101101101110 = \gamma^{455} = \beta^7, y_2 = 111110010000 = \gamma^{195} = \beta^3, \text{ where } \beta \text{ is a primitive element of } GF(2^6).$$

$$\underline{G} = \begin{bmatrix} 111110010000 \\ 101101101110 \\ 100011010111 \end{bmatrix}$$

The coefficients are given by

$$\begin{bmatrix} a_6 \\ a_5 \\ a_3 \end{bmatrix} = \begin{bmatrix} \gamma^{2340} & \gamma^{1755} & \gamma^{2925} \\ \gamma^{585} & \gamma^{3510} & \gamma^{1755} \\ \gamma^{1170} & \gamma^{2925} & \gamma^{3510} \end{bmatrix} \begin{bmatrix} \gamma^{715} \\ \gamma^{455} \\ \gamma^{195} \end{bmatrix} = \begin{bmatrix} \gamma^{1755} \\ \gamma^{3705} \\ \gamma^{1040} \end{bmatrix}$$

$$\text{Thus } f_3(x) = \gamma^{1040} x^4 + \gamma^{3705} x^2 + \gamma^{1755} x.$$

We see that $\gamma^{1040} = \beta^{16}$, $\gamma^{3705} = \beta^{57}$ and $\gamma^{1755} = \beta^{27}$ belong to $\text{GF}(2^6)$ and since $3|6$, the coefficients are independent of each other.

Example 3.8.9: Finally, we consider an example of a linear (n,k) code in which all the code vectors $\in \text{GF}(2^{n_1})$ where $k \nmid n_1$. Let $n = 12$, $k = 3$ and $n_1 = 4$. Let α , β and γ be primitive elements of $\text{GF}(2^3)$, $\text{GF}(2^4)$ and $\text{GF}(2^{12})$ respectively.

The minimal polynomials for generating $\text{GF}(2^3)$ and $\text{GF}(2^{12})$ are chosen to be same as in the previous example. b_{ij} 's also have been calculated.

$$y_0 = 010111110110 = \gamma^{1911} = \beta^7, y_1 = 010010011111 = \gamma^{546} = \beta^2, y_2 = 000101101000 = \gamma^{273} = \beta, \text{ where } \beta \text{ is a primitive element of } \text{GF}(2^4).$$

$$\underline{G} = \begin{bmatrix} 000101101000 \\ 010010011111 \\ 010111110110 \end{bmatrix}$$

The coefficients are given by

$$\begin{bmatrix} a_6 \\ a_5 \\ a_3 \end{bmatrix} = \begin{bmatrix} \gamma^{2340} & \gamma^{1755} & \gamma^{2925} \\ \gamma^{585} & \gamma^{3510} & \gamma^{1755} \\ \gamma^{1170} & \gamma^{2925} & \gamma^{3510} \end{bmatrix} \begin{bmatrix} \gamma^{1911} \\ \gamma^{546} \\ \gamma^{273} \end{bmatrix} = \begin{bmatrix} \gamma^{2019} \\ \gamma^{3639} \\ \gamma^{894} \end{bmatrix}$$

$$\text{Thus } f_8(x) = \gamma^{894} x^4 + \gamma^{3639} x^2 + \gamma^{2019} x,$$

which can be expressed as a single term LFP as

$$f_8(x) = \text{frs}(\gamma^{2019} x),$$

$$\text{where } \text{frs}(\Theta) = \Theta + \Theta^{16} + \Theta^{256}.$$

The coefficients belong to $\text{GF}(2^{12})$ and satisfy conjugacy relations given by

$$(a_6)^{16} = a_5, (a_5)^{16} = a_3, (a_3)^{16} = a_6 \text{ (modulo 4095)}.$$

In the following sections, we study the algebraic properties of *single term* LGPs:

3.9 Algebraic Structures of Single Term Linearized GPs

3.9.1 Group Structure of GPs of the form

$$\beta^j f(x), j = -\infty, 0, \dots, 2^n - 2.$$

The following theorem is about the structure of the set of any GSFs (not necessarily linearized) represented by LGPs of the form

$$\beta^j f(x) = \sum_{i=0}^{k-1} \beta^i \gamma_i x^{2^i}, j = -\infty, 0, 1, 2, \dots, 2^n - 2, \quad (3.9.1)$$

where $f(x)$ represents any mapping from $\text{GF}(2^k)$ to $\text{GF}(2^n)$.

Theorem 3.9.1: The set of GSFs mapping from $\text{GF}(2^k)$ to $\text{GF}(2^n)$, with their GP coefficients from $\text{GF}(2^L)$, $L = \text{L.C.M. of } n \text{ and } k$, of the form (3.9.1) has the structure of an *additive abelian group*.

Proof: It is sufficient to prove the closure property of this set under addition.

Let $\beta^{h_1} f(x)$ and $\beta^{h_2} f(x)$ be two elements of the set. Then $\beta^{h_1} f(x) + \beta^{h_2} f(x) = \beta^{h_3} f(x)$ (say), where $\beta^{h_3} = \beta^{h_1} + \beta^{h_2}$, is also a member of the same set, since β^{h_3} also belongs to $\text{GF}(2^n)$.

Other axioms of the additive abelian group structure may be easily seen to be satisfied in this case. Q.E.D.

Sets of single term LGPs of the form (3.9.1) also obviously satisfy the group structure given in Theorem 3.9.1.

3.9.2 Algebraic Structure of Single Term Linearized Polynomials

In this subsection, we discuss about the additional structure possessed by the class of single term LPs of the form $\beta^j x$, $j = -\infty, 0, 1, \dots, 2^n-2$, besides the group structure. Single term LPs represent linear mappings from $GF(2^k)$ to $GF(2^n)$, only when $k|n$, the coefficients also thus belonging to $GF(2^n)$. In Chapter 5, we will prove that the corresponding LFs always represent one-to-one mappings and hence linear (n,k) codes.

In Theorem 3.9.2, we show that the set of single term LPs has the structure of a finite field (F_p):

Theorem 3.9.2: The set of single term LPs mapping from $GF(2^k)$ to $GF(2^n)$, where $k|n$, with coefficients from $GF(2^n)$, of the form

$$\beta^j x, j = -\infty, 0, 1, 2, \dots, 2^n-2,$$

β being a primitive element of $GF(2^n)$, has the structure of a finite field F_p , isomorphic to $GF(2^n)$, under the operations of addition and symbolic multiplication.

Proof: That this set is an abelian group under addition, has been proved in Theorem 3.9.1.

Let us consider the second operation. We define the operation of symbolic multiplication on the nonzero elements of the set. Let us consider two nonzero elements of the set, say,

$f_1(x) = \beta^{h_1} x$ and $f_2(x) = \beta^{h_2} x$. Then the symbolic multiplication of $f_1(x)$ and $f_2(x)$ given by

$$f_1(x) (x) f_2(x) = f_1(f_2(x)) = \beta^{h_1} (\beta^{h_2} x) = \beta^{h_1+h_2} x,$$

is also a member of the same set, since $\beta^{h_1+h_2}$ also belongs to $GF(2^n)$. We also see that the

operation is commutative in this case, although symbolic multiplication, in general, is noncommutative. The identity element may be seen to be equal to $\beta^0 x$, and the inverse of an element $\beta^{h_1} x$ is equal to $\beta^{-h_1} x$, where $-h_1$ is taken modulo $2^n - 1$. Further the set is associative.

Finally the symbolic multiplication distributes over addition.

Thus the set has the structure of a finite field of order 2^n . Since fields of a given order are isomorphic, this finite field is isomorphic to $GF(2^n)$ Q.E.D.

3.9.3 Algebraic Structure of Single Term Linearized Frobenius Polynomials

Similar to single term LPs, single term LFPs also possess the structure of a finite field. For describing this structure we define an operation of composition between two single term LFPs. We denote this operation as *Frobenius symbolic multiplication*. This operation is, in general, noncommutative.

3.9.3.1 Frobenius Symbolic Multiplication

Definition 3.9.1: Let $f_1(x)$ and $f_2(x)$ be two single term LFPs. Then Frobenius symbolic multiplication of $f_1(x)$ and $f_2(x)$ is defined as

$$f_1(x) (x) f_2(x) = f_1(f_2(x)). \quad (3.9.2a)$$

whereas Frobenius symbolic multiplication of $f_2(x)$ and $f_1(x)$ is defined as

$$f_2(x) (x) f_1(x) = f_2(f_1(x)). \quad (3.9.2b)$$

In general, $f_1(f_2(x)) \neq f_2(f_1(x))$.

Theorem 3.9.3: Frobenius symbolic multiplication of two single term LFPs, say $f_1(x) = \text{frs}(\gamma^j x)$ and $f_2(x) = \text{frs}(\gamma^i x)$, gives another single term LFP, say, $f_3(x) = \text{frs}(\gamma^s x)$, where

$$\gamma^s = \gamma^j \sum_{m=0}^{t-1} \gamma^i Q^m, \quad (3.9.3)$$

and $f_1(x)$, $f_2(x)$ and $f_3(x)$ represent linear mappings from $GF(2^k)$ to $GF(2^n)$, the coefficients γ^i , γ^j and γ^s belonging to $GF(2^L)$, L being the L.C.M. of n and k , and $t = L/n$.

roof: We have

$$\text{frs}(\Theta) = \Theta + \Theta^Q + \Theta^{Q^2} + \dots + \Theta^{Q^{t-1}}; \Theta^{Q^t} = \Theta,$$

where $Q = 2^n$, and $t = L/n$.

$$\text{Therefore } f_1(x) = \text{frs}(\gamma^i x) = \gamma^i x + (\gamma^i x)^Q + (\gamma^i x)^{Q^2} + \dots + (\gamma^i x)^{Q^{t-1}}.$$

$$\text{Similarly, } f_2(x) = \text{frs}(\gamma^j x) = \gamma^j x + (\gamma^j x)^Q + (\gamma^j x)^{Q^2} + \dots + (\gamma^j x)^{Q^{t-1}}.$$

$$\text{Thus } \text{frs}(\gamma^i x) (x) \text{frs}(\gamma^j x) = \text{frs}(\gamma^i (\text{frs}(\gamma^j x)))$$

$$= \text{frs}(\gamma^i (\gamma^j x + (\gamma^j x)^Q + (\gamma^j x)^{Q^2} + \dots + (\gamma^j x)^{Q^{t-1}}))$$

$$= \text{frs}(\gamma^{i+j} x) + \text{frs}(\gamma^{i+jQ} x^Q) + \text{frs}(\gamma^{i+jQ^2} x^{Q^2}) + \dots + \text{frs}(\gamma^{i+jQ^{t-1}} x^{Q^{t-1}}).$$

$$\text{Now, } \text{frs}(\gamma^{i+jQ} x^Q) = \gamma^{i+jQ} x^Q + \gamma^{(i+jQ)Q} x^{Q^2} + \gamma^{(i+jQ)Q^2} x^{Q^3} + \dots + \gamma^{(i+jQ)Q^{t-1}} x^{Q^{t-1}}$$

$$= \text{frs}(\gamma^{(i+jQ)Q^{t-1}} x^{Q^{t-1}}) x.$$

$$\text{Similarly, } \text{frs}(\gamma^{i+jQ^2} x^{Q^2}) = \gamma^{i+jQ^2} x^{Q^2} + \gamma^{(i+jQ^2)Q} x^{Q^3} + \dots + \gamma^{(i+jQ^2)Q^{t-2}} x^{Q^{t-2}} + \gamma^{(i+jQ^2)Q^{t-1}} x^{Q^{t-1}}$$

$$= \text{frs}(\gamma^{(i+jQ^2)Q^{t-2}} x^{Q^{t-2}})$$

Finally,

$$\text{frs}(\gamma^{i+jQ^{t-1}} x^{Q^{t-1}}) = \gamma^{i+jQ^{t-1}} x^{Q^{t-1}} + \gamma^{(i+jQ^{t-1})Q} x^{Q^t} + \dots + \gamma^{(i+jQ^{t-1})Q^{t-1}} x^{Q^{t-1}}$$

$$= \text{frs}(\gamma^{(i+jQ^{t-1})Q} x^{Q^{t-1}}).$$

$$\text{Thus } \text{frs}(\gamma^i x) (x) \text{frs}(\gamma^j x) = \text{frs}(\gamma^{i+j} x) + \text{frs}(\gamma^{(i+jQ)Q^{t-1}} x^{Q^{t-1}}) + \text{frs}(\gamma^{(i+jQ^2)Q^{t-2}} x^{Q^{t-2}}) + \dots +$$

$$\text{frs}(\gamma^{(i+jQ^{t-1})Q} x^{Q^{t-1}}) = \text{frs}([\gamma^{i+j} + \gamma^{(i+jQ)Q^{t-1}} + \gamma^{(i+jQ^2)Q^{t-2}} + \dots + \gamma^{(i+jQ^{t-1})Q}] x)$$

$$= \text{frs}((\gamma^{i+j} + \gamma^{iQ^{t-1}+j} + \gamma^{iQ^{t-2}+j} + \dots + \gamma^{iQ+j})x), \text{ (where } Q^t = 1 \text{ modulo } L-1).$$

$$= \text{frs}(\gamma^j(\gamma^i + \gamma^{iQ} + \gamma^{iQ^2} + \dots + \gamma^{iQ^{t-2}} + \gamma^{iQ^{t-1}})x)$$

$$= \text{frs}(\gamma^j(\sum_{m=0}^{t-1} \gamma^{iQ^m})x). \quad \text{Q.E.D.}$$

Corollary 3.9.1: Frobenius symbolic multiplication is *commutative* if the exponents i and j in the coefficients γ^i and γ^j of the LFPs $\text{frs}(\gamma^i x)$ and $\text{frs}(\gamma^j x)$ satisfy the relation

$$i = (j + h\nu) \text{ modulo } 2^L-1, \quad (3.9.4)$$

where $\nu = (2^L-1)/(2^n-1)$ and $h = 0, 1, 2, \dots, 2^n-2$.

Proof: We have $\text{frs}(\gamma^i x)(x) \text{frs}(\gamma^j x) = \text{frs}(\gamma^s x)$, where $\gamma^s = \gamma^j \sum_{m=0}^{t-1} \gamma^{iQ^m}$,

let $i = j + h\nu$. Now, since $\text{GF}(2^n)$ is a subfield of $\text{GF}(2^L)$, the nonzero elements of $\text{GF}(2^n)$ expressed as powers of the primitive element γ of $\text{GF}(2^L)$ is $\gamma^{h\nu}$, $h = 0, 1, 2, \dots, 2^n-2$.

Thus $h\nu$ satisfies $(h\nu)^{Q^m} = h\nu$.

$$\text{Therefore } \gamma^s = \gamma^j \sum_{m=0}^{t-1} \gamma^{(j+h\nu)Q^m} = \gamma^{j+h\nu} \sum_{m=0}^{t-1} \gamma^{jQ^m} = \gamma^j \sum_{m=0}^{t-1} \gamma^{jQ^m}.$$

$$\text{frs}(\gamma^j x)(x) \text{frs}(\gamma^j x). \quad \text{Q.E.D.}$$

3.9.3.2 Finite Field Structure

We note from the result on the commutativity of Frobenius symbolic multiplication given in Corollary 3.9.1, that we can form cyclic groups of order 2^n-1 under Frobenius symbolic multiplication with the members of the group given by $\text{frs}(\gamma^{j+h\nu} x) = \text{frs}(\beta^h \gamma^j x)$, $h = 0, 1, 2, \dots, 2^n-2$, where β is a primitive element of $\text{GF}(2^n)$. Further, if we include the function $\text{frs}(\beta^{-\infty} \gamma^j x) = 0$, to the above set, then we can prove that the set of LFPs $\text{frs}(\beta^h \gamma^j x)$, $h = -\infty, 0, 1, 2, \dots, 2^n-2$, has the structure of a finite field (F_p) isomorphic to

$\mathbb{F}(2^n)$, as stated and proved in the next theorem.

Theorem 3.9.4: The set of single term LFPs of the form $\text{frs}(\beta^h \gamma^j x)$, $h = -\infty, 0, 1, 2, \dots, 2^n-2$, where γ and β are primitive elements of $\text{GF}(2^L)$ and $\text{GF}(2^n)$ respectively, ($\text{GF}(2^n)$ being a subfield of $\text{GF}(2^L)$) and γ^j satisfies

$$\text{frs}(\gamma^j) = \sum_{m=0}^{t-1} \gamma^{jQ^m} = 1, \quad (3.9.5)$$

$= L/n$, has the structure of a finite field F_f isomorphic to $\text{GF}(2^n)$ under the operations of addition and Frobenius symbolic multiplication.

The multiplicative identity element of the field is $\text{frs}(\beta^0 \gamma^j x) = \text{frs}(\gamma^j x)$, which is also an idempotent element.

The multiplicative inverse of an element $\text{frs}(\beta^h \gamma^j x)$, $h = 0, 1, 2, \dots, 2^n-2$, is equal to $\text{frs}(\beta^{-h} \gamma^j x)$ where β^{-h} is taken modulo 2^n-1 .

Proof: First we prove the closure of the set $\text{frs}(\beta^h \gamma^j x)$, $h = -\infty, 0, 1, \dots, 2^n-2$, under the addition operation:

Let $\text{frs}(\beta^{h_1} \gamma^j x)$ and $\text{frs}(\beta^{h_2} \gamma^j x)$ be two elements of the set. Then

$\text{frs}(\beta^{h_1} \gamma^j x) + \text{frs}(\beta^{h_2} \gamma^j x) = \text{frs}((\beta^{h_1} + \beta^{h_2}) \gamma^j x) = \text{frs}(\beta^{h_3} \gamma^j x)$ (say), is again a member of the same set. This is because since β^{h_1}, β^{h_2} belongs to $\text{GF}(2^n)$, their sum $\beta^{h_1} + \beta^{h_2} = \beta^{h_3}$ (say), also belongs to $\text{GF}(2^n)$.

It may be easily verified that the other axioms of the additive abelian group structure, namely, commutativity, associativity, presence of additive identity element (being '0') and inverse, are satisfied. So we next consider the second operation, i.e., Frobenius symbolic multiplication, and prove that under this operation, the nonzero elements of this set form a cyclic group of order 2^n-1 .

and the

1) **Closure:** Let $\text{frs}(\beta^{h_1} \gamma^j x)$ and $\text{frs}(\beta^{h_2} \gamma^j x)$ be two elements of the set.

Then as per Theorem 3.9.3, we have

$$\text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_2} \gamma^j x) = \text{frs}((\beta^{h_1+h_2} \gamma^j \sum_{m=0}^{t-1} \gamma^{jQ^m})x).$$

Now, $\sum_{m=0}^{t-1} \gamma^{jQ^m} = \text{frs}(\gamma^j) \in \text{GF}(2^n)$. So let $\text{frs}(\gamma^j) = \beta^{h_3}$ (say).

Thus $\text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_2} \gamma^j x) = \text{frs}((\beta^{h_1+h_2+h_3} \gamma^j x) = \text{frs}(\beta^{h_4} \gamma^j x)$ (say), is again a member of the same set. Further, we note from Corollary 3.9.1, that this operation is commutative.

We will later on prove that $\text{frs}(\gamma^j) = \beta^{h_3} = 1$, so that

$$\text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_2} \gamma^j x) = \text{frs}(\beta^{h_2} \gamma^j x) (x) \text{frs}(\beta^{h_1} \gamma^j x) = \text{frs}(\beta^{h_1+h_2} \gamma^j x). \quad (3.9.6)$$

2) **Associativity:** Let $f_1 = \text{frs}(\beta^{h_1} \gamma^j x)$, $f_2 = \text{frs}(\beta^{h_2} \gamma^j x)$ and $f_3 = \text{frs}(\beta^{h_3} \gamma^j x)$ be three elements of the set. Then for associativity, we require,

$$\begin{aligned} (f_1 (x) f_2) (x) f_3 &= f_1 (x) (f_2 (x) f_3) \\ f_1 (x) f_2 &= \text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_2} \gamma^j x) = \text{frs}((\beta^{h_1+h_2} \gamma^j x). \\ f_2 (x) f_3 &= \text{frs}(\beta^{h_2} \gamma^j x) (x) \text{frs}(\beta^{h_3} \gamma^j x) = \text{frs}((\beta^{h_2+h_3} \gamma^j x). \\ (f_1 (x) f_2) (x) f_3 &= \text{frs}(\beta^{h_1+h_2} \gamma^j x) (x) \text{frs}(\beta^{h_3} \gamma^j x) = \text{frs}((\beta^{h_1+h_2+h_3} \gamma^j x). \end{aligned}$$

Similarly,

$$\begin{aligned} f_1 (x) (f_2 (x) f_3) &= \text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}((\beta^{h_2+h_3} \gamma^j x) \\ &= \text{frs}((\beta^{h_1+h_2+h_3} \gamma^j x). \end{aligned}$$

Thus the operation is associative.

(3) **Identity:** We prove that the identity element in the field is $\text{frs}(\beta^0 \gamma^j x) = \text{frs}(\gamma^j x)$

and that $\text{frs}(\gamma^j) = \sum_{m=0}^{t-1} \gamma^{jQ^m} = 1$. We also prove that $\text{frs}(\gamma^j x)$ is an idempotent element,

$$\text{frs}(\gamma^j x) (x) \text{frs}(\gamma^j x) = \text{frs}(\gamma^j x).$$

Let $f_1 = \text{frs}(\beta^{h_1} \gamma^j x)$ be any element in the set and let $f_e = \text{frs}(\beta^{h_e} \gamma^j x)$ be the identity element. Then according to Theorem 3.9.3, we should have

$$\begin{aligned} \text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_e} \gamma^j x) &= \text{frs}(\beta^{h_e} \gamma^j x) (x) \text{frs}(\beta^{h_1} \gamma^j x) = \text{frs}((\beta^{h_1+h_e} \gamma^j \sum_{m=0}^{t-1} \gamma^{jQ^m}) x) \\ &= \text{frs}(\beta^{h_1} \gamma^j x), \text{ in which case } \beta^{h_e} \text{ and } \sum_{m=0}^{t-1} \gamma^{jQ^m} \text{ should both be equal to 1.} \end{aligned}$$

Thus the identity element is $f_e = \text{frs}(\gamma^j x)$, where γ^j satisfies

$$\text{frs}(\gamma^j) = \sum_{m=0}^{t-1} \gamma^{jQ^m} = 1.$$

Now let us compute $\text{frs}(\gamma^j x) (x) \text{frs}(\gamma^j x)$, which can be written as,

$$\text{frs}(\beta^0 \gamma^j x) (x) \text{frs}(\beta^0 \gamma^j x) = \text{frs}((\beta^{0+0} \gamma^j \sum_{m=0}^{t-1} \gamma^{jQ^m}) x) = \text{frs}(\beta^0 \gamma^j x), \text{ since } \sum_{m=0}^{t-1} \gamma^{jQ^m} = 1.$$

Thus we get $\text{frs}(\gamma^j x) (x) \text{frs}(\gamma^j x) = \text{frs}(\gamma^j x)$, meaning that $\text{frs}(\gamma^j x)$ satisfies the idempotent property.

(4) **Inverse:** We prove that the inverse of an element $\text{frs}(\beta^{h_1} \gamma^j x)$ in the set is

$$\text{frs}(\beta^{-h_1} \gamma^j x) \text{ where } -h_1 \text{ is taken modulo } 2^n - 1.$$

Let $f_1 = \text{frs}(\beta^{h_1} \gamma^j x)$ be the inverse of an element $f_1 = \text{frs}(\beta^{h_1} \gamma^j x)$, and let $f_e = \text{frs}(\gamma^j x)$ be the identity element. Then we should have

$$f_1 (x) f_1 = f_e.$$

$$\text{ie., } \text{frs}(\beta^{h_1} \gamma^j x) (x) \text{frs}(\beta^{h_1} \gamma^j x) = \text{frs}((\beta^{h_1+h_1} \gamma^j x) = \text{frs}(\gamma^j x),$$

which means, we should have $h_1 = -h_1$ (modulo $2^n - 1$).

Further, Frobenius symbolic multiplication distributes over addition. Thus F_f has the structure of a finite field of order 2^n . Since finite fields of the same order are isomorphic, F_f is isomorphic to $\text{GF}(2^n)$.

Before concluding this chapter, we give an example to illustrate the above concepts:

Example 3.9.1: In this example, we consider only those LFFs which generate one-to-one mappings (ie., linear (n,k) codes).

Let $n = 3$, $k = 2$. All the (3,2) linear codes can be represented by single term LFPs, G.C.D. of 3 and 2 = 1. The number of linear transformations in this case is $n = (2^3 - 1)(2^3 - 2) = 42$. The coefficients of the LFPs belong to $GF(2^6)$. Let $x^6 + x + 1$ be primitive polynomial for generating this field, with γ as a primitive element. Then the fields $GF(2^3)$ and $GF(2^2)$ are generated by the primitive polynomials $x^3 + x^2 + 1$ and $x^2 + x + 1$ respectively, with primitive elements β and α respectively.

Let us now choose those elements γ^j of $GF(2^6)$ whose Frobenius sum with respect to $GF(2^3)$ is 1, ie., $\sum_{m=0}^1 \gamma^{jQ^m} = 1$. There are 8 values in $GF(2^6)$ whose Frobenius sum with respect to $GF(2^3)$ is 1. They are γ^j , $j = 11, 21, 22, 25, 37, 42, 44$ and 50. Out of these, the functions $\text{frs}(\gamma^j x)$, $j = 21$ and 42 do not generate one-to-one mappings (the condition for an LFP to represent a one-to-one mapping will be derived in Chapter 5). Thus we choose the remaining 6 elements as idempotents/identities (since we are interested only in one-to-one mappings in this example), and all the 42 linear transformations are grouped into 6 finite fields of order 8, each of them isomorphic to $GF(2^3)$. The nonzero LFPs in each field would represent a (3,2) code. We take one of the above field and form the Frobenius symbolic multiplication, and addition tables (Cayley Tables) in Table 3.1, to illustrate Theorem 3.9.4.

Let the identity element chosen be $\text{frs}(\gamma^{11} x)$. Then the remaining nonzero members of the finite field are $\text{frs}(\gamma^j x)$, $j = 20, 29, 38, 47, 56$ and 2, as $\beta = \gamma^9$.

Only the exponents of γ are listed in Table 3.1.

Sample Computation: Let us compute $\text{frs}(\gamma^{20} x)(x) \text{frs}(\gamma^{29} x)$. This may be written as $\text{frs}(\gamma^9 \gamma^{11} x)(x) \text{frs}(\gamma^{18} \gamma^{11} x) = \text{frs}(\beta \gamma^{11} x)(x) \text{frs}(\beta^2 \gamma^{11} x) = \text{frs}(\beta^3 \gamma^{11} x) = \text{frs}(\gamma^{38} x)$ according to (3.9.6), where $\text{frs}(\gamma^{11} x)$ is the identity.

milarly, inverse of the element $\text{frs}(\gamma^{20} x)$ may be computed as the inverse of $\text{frs}(\beta \gamma^{11} x)$
 $\text{frs}(\beta^{-1} \gamma^{11}) = \text{frs}(\beta^6 \gamma^{11}) = \text{frs}(\gamma^2 x)$.

**Table 3.1: Cayley Tables for the Finite Field F_f Comprising of Single
Term Linearized Frobenius Polynomials Representing
Linear (3,2) Codes**

(a) Frobenius Symbolic Multiplication Table

(x)	11	20	29	38	47	56	2
11	11	20	29	38	47	56	2
20	20	29	38	47	56	2	11
29	29	38	47	56	2	11	20
38	38	47	56	2	11	20	29
47	47	56	2	11	20	29	38
56	56	2	11	20	29	38	47
2	2	11	20	29	38	47	56

(b) Addition Table

+	11	20	29	38	47	56	2
11	$-\infty$	56	38	29	2	20	47
20	56	$-\infty$	2	47	38	11	29
29	38	2	$-\infty$	11	56	47	20
38	29	47	11	$-\infty$	20	2	56
47	2	38	56	20	$-\infty$	29	11
56	20	11	47	2	29	$-\infty$	38
2	47	29	20	56	11	38	$-\infty$

CHAPTER 4

GSF THEORY FOR BOOLEAN FUNCTIONS

In this chapter, we consider the theory of GSFs as applied to the class of Boolean functions (BFs). We show that k -variable BFs can be represented by Frobenius functions (Fs). The monoid algebra structure of GSFs discussed in Chapter 2 is applied to the class BFs and the class of linear Boolean functions (LBFs) are described as ideals in this algebra. This is then extended to the class of generalized Reed-Muller (GRM) codes which are constructed from LBFs. It is shown that any r^{th} order GRM code may be viewed as an ideal in a monoid algebra.

Another topic which we examine is the one on equivalence relations used to classify BFs and their effect on the GP coefficients of the corresponding BFs. Consequently, a finite field model which implements various operations corresponding to the equivalence relations proposed for the synthesis of BFs. Alternatively, a finite field model based on the Frobenius sum representation of a BF is suggested. We also propose some new equivalence relations which are a consequence of the monoid algebra structure of BFs, and an attempt is made to use them to classify BFs. Lastly we characterize classes of self-dual (SD) and anti self-dual (ASD) BFs using GSFs.

It may be noted that in all our discussions the coefficients of the GP representing a BF are in fact the Galois Transform (GT) coefficients of a vector of length 2^k over $GF(2)$.

4.12 Representation of Boolean Functions by GPs

In this section, we show that any k -variable BF may be represented by an appropriate GP with coefficients from $GF(2^k)$. In representing BFs using GPs, we assume

that the BF components are in the order of the power of a primitive element γ in $GF(2^k)$, i.e., in the order $\gamma^{-\infty}, \gamma^0, \gamma, \dots, \gamma^{2^k-2}$. It may be recalled that we denoted this order as the *field order* in Chapter 2. This is in contrast to the conventional ordering which is called the *natural order* since in this case, the components of a BF are taken in the order 0, 1, 2, ..., 2^k-1 .

A k -variable BF is a mapping from $GF(2^k)$ to $GF(2)$. Thus it may be represented by a GP. The coefficients satisfy conjugacy constraints, and therefore it may be represented in general by a *Frobenius polynomial (FP)*. This is stated in the following theorem:

Theorem 4.1.1: Any k -variable BF may be represented by a Frobenius polynomial (FP) of the form

$$f(x) = a_{-\infty} + \sum_j \text{frs}(a_j x^{-j}), \quad (4.1.1)$$

where j is one member of a conjugacy class modulo 2^k-1 , and $-j$ is taken modulo 2^k-1 .

Further, the coefficients $a_{-\infty}$ and a_0 belong to $GF(2)$.

The coefficients are given by (2.3.13).

Proof: Since a BF is a mapping from $GF(2^k)$ to $GF(2)$, its coefficients lie in $GF(2^k)$. Conjugacy relations will always exist among the coefficients. Hence the function may be represented by the sum of various Frobenius terms, i.e., a FP. The constant term

$a_{-\infty} = f(\gamma^{-\infty})$, and therefore belongs to $GF(2)$. a_0 is the sum of all the function values. It belongs to $GF(2)$, since all the function values belong to $GF(2)$ Q.E.D.

4.2 Monoid Algebra Model of Boolean Functions

Since BFs are mappings from $GF(2^k)$ to $GF(2)$, we may associate them with the elements of a monoid algebra over $GF(2)$. Viewing the class of BFs as a monoid algebra

er $GF(2)$, helps in the algebraic characterization of LBFs and GRM codes. Further, the binary operations in this algebra may as well be used as equivalence relations for classification of BFs. A study of the ideals in this algebra is helpful in the above cases. Hence we take up this task in the next subsection.

2.1 Ideals in the Monoid Algebra of Boolean Functions

In Chapter 2, we have seen that a FP can be expressed as a sum of elements from minimal ideals in a monoid algebra. Further, each of these minimal ideals are obtained by signing all the values of $GF(Q^d)$ to one Frobenius class at a time (where d is the order of a Frobenius class, and $Q = 2$ for BFs), and forcing the elements of the remaining Frobenius classes to zero. The number of minimal ideals in this algebra is equal to the number of Frobenius classes given by n_{frob} of (2.4.3) and the number of ideals is equal to n_{frob} , as any ideal in this algebra may be expressed as a direct sum of ' n_{frob} ' minimal ideals.

In Table 4.1, we list the minimal ideals in the case of 2 and 3 variable BFs. The GP coefficients in the order $a_{-\infty}, a_0, a_1, \dots, a_\xi$ (where $\xi = 2^k - 2$), are listed as a power of a primitive element γ in $GF(2^k)$. Only the exponents of γ are listed.

a) $k = 2$

In this case, the number of minimal ideals = $n_{frob} = 3$. They are listed in Table 4.1a. By taking direct sum of the above minimal ideals we may get a total of 8 ideals in this algebra.

b) $k = 3$

In this case the number of minimal ideals = $n_{frob} = 4$. They are listed in Table 4.1b. We have a total of 16 ideals in this algebra.

Table 4.1 Minimal Ideals in a Monoid Algebra Consisting
of k -Variable Boolean Functions

(a) $k = 2$

	$a_{-\infty}$	a_0	a_1	a_2
I1	$-\infty$ 0	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$
I2	$-\infty$ $-\infty$	$-\infty$ 0	$-\infty$ $-\infty$	$-\infty$ $-\infty$
I3	$-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ 0 2 1	$-\infty$ 0 1 2

(b) $k = 3$

	$a_{-\infty}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
I1	$-\infty$ 0	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$
I2	$-\infty$ $-\infty$	$-\infty$ 0	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$	$-\infty$ $-\infty$
I3	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ 0 2 4 6 1 3 5	$-\infty$ 0 4 1 5 2 6 3	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ 0 1 2 3 4 5 6	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$
I4	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ 0 4 1 5 2 6 3	$-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$ $-\infty$	$-\infty$ 0 2 4 6 1 3 5	$-\infty$ 0 1 2 3 4 5 6

3 Algebraic Characterization of Linear Boolean Functions

In this section, we characterize linear Boolean functions (LBFs) using GSFs and discuss their algebraic structure.

Definition 4.3.1: A LBF is of the form

$$f(x) = \sum_{i=1}^k \ell_i x_i, \quad (4.3.1)$$

where $x = x_1 x_2 \dots x_k$, and $x_i, \ell_i \in GF(2)$ $i = 1, 2, \dots, k$.

The number of LBFs of k variables $= 2^k$.

3.1 Representation of Linear Boolean Functions by Linearized GPs

Theorem 4.3.1: Any LBF of k variables may be represented by a single term linearized Frobenius polynomial (LFP) of the form

$$f(x) = \text{frs}(\gamma^i x), \quad i = -\infty, 0, 1, \dots, 2^k-2, \quad (4.3.2)$$

where γ is a primitive element of $GF(2^k)$.

Proof: LBFs have the property that $f(y_1 + y_2) = f(y_1) + f(y_2)$ where y_1 and y_2 are input tuples. In Chapter 3, we have seen that such GPs have a linearized Galois polynomial (GP) representation. Further, since the mapping is from $GF(2^k)$ to $GF(2)$, the coefficients belong to $GF(2^k)$, and conjugacy relations exist among the coefficients, allowing us to present them by single term LFPs, $\text{frs}(\gamma^i x)$, $i = -\infty, 0, 1, \dots, 2^k-2$. We get 2^k functions, corresponding to each element of $GF(2^k)$. Thus LBFs may be represented by single term LFPs. Q.E.D.

From Theorem 4.3.1, an alternative description of LBFs in terms of its Galois

pectrum or GT coefficients may be formed as follows:

A LBF is a signal vector of length 2^k over $GF(2)$ whose Galois spectrum, L_i , $i = -\infty, 1, \dots, 2^k-2$, is identically zero except in those indices -2^j , $j = 0, 1, \dots, k-1$, with the spectral coefficients satisfying conjugacy constraints.

3.2 Linear Boolean Functions as Ideals in a Monoid Algebra

The class of LBFs exhibit interesting algebraic properties. The fact that this class is a group structure under pointwise addition, is already known. However, we note that viewing them in the framework of a monoid algebra allows one to see its inherent additional algebraic properties. We investigate these properties in this subsection.

We state the following theorem giving the algebraic structure of LBFs.

Theorem 4.3.2: The class of LBFs is an *ideal* in the cyclic monoid algebra over $GF(2)$ consisting of k -variable BFs.

Proof: Let us take up the two binary operations in a monoid algebra consisting of variable BFs in which the LBFs under consideration form a subclass.

It is known that the LBF class forms a group under pointwise addition [16].

So let us consider the convolution operation as defined in Chapter 3. If we take any BF, say r , belonging to the monoid algebra, and convolve with any function, say ℓ , in the LBF class, we get a function within the LBF class. This may be better understood by considering the corresponding operation of pointwise multiplication on the GP coefficients. If the GP coefficients of $r(x)$ be denoted as

$$R_{-\infty}, R_0, R_1, \dots, R_{\xi}; \xi = 2^k-2,$$

and those of $\ell(x)$ be denoted as

$$L_{-\infty}, L_{-2^j}, j = k-1, k-2, \dots, 1, 0,$$

here $L_{-\infty} = 0$, and -2^j is taken modulo $2^k - 1$.

Now we see that pointwise multiplication of the GP coefficients of $r(x)$ and $l(x)$ reduces all the coefficients of the resulting function to zero except those with indices -2^j , $j = 1, \dots, k-1$. In other words, we get the resulting function as a LBF. Thus we see that the class of LBFs forms an ideal in the cyclic monoid algebra consisting of k -variable BFs.

Q.E.D.

Example 4.3.1: In Table 4.1 listed earlier, I3 of 2-variables and I4 of 3-variables respectively represent 2-variable and 3-variable LBFs.

In the next section, we deal with the class of generalized Reed-Muller (GRM) codes which are constructed from LBFs and show that they can be viewed as ideals in a monoid algebra.

4 Algebraic Characterization of Generalized Reed-Muller Codes

The fact that Reed-Muller (RM) codes contain code vectors of length 2^k allows us to characterize these codes using GSFs, since Galois Transform operates on vectors of length 2^k . For this study, we take up the class of Generalized Reed-Muller (GRM) codes over $GF(2)$. This is because GRM codes contain code vectors whose components are taken in the field order rather than the original RM codes whose code vector components are taken in the natural order. Since GRM codes are formed from LBFs, this allows us to extend the algebraic structures defined for LBFs in Section 4.3, to GRM codes also.

We define the original RM codes first to get a clearer understanding of the more general class of GRM codes.

Definition 4.4.1: For every integer k and r , where $r < k$, there exists a RM code of block length 2^k called the r^{th} order RM code of block length 2^k , defined by a generator matrix G_{RM} as follows:

$$G_{\text{RM}} = \begin{bmatrix} G_{\text{RM}_0} \\ G_{\text{RM}_1} \\ \vdots \\ G_{\text{RM}_r} \end{bmatrix}, \quad (4.4.1)$$

where G_{RM_0} is a 2^k length vector containing all ones, G_{RM_1} is a $k \times 2^k$ matrix which contains as its columns all binary k -tuples in the natural order $0, 1, 2, \dots, 2^k-1$, and any G_{RM_s} , $2 \leq s \leq r$, is obtained from G_{RM_1} by taking its rows to be all possible componentwise products of the rows of G_{RM_1} taken s at a time, where the componentwise products of two vectors \underline{x} and \underline{y} given by

$$\underline{x} = \{x_0, x_1, x_2, \dots, x_\rho\}$$

and $\underline{y} = \{y_0, y_1, y_2, \dots, y_\rho\}$

is defined as

$$\underline{x} \cdot \underline{y} = \{x_0 y_0, x_1 y_1, \dots, x_\rho y_\rho\}, \quad (4.4.2)$$

where $\rho = 2^k - 1$.

Since there are $\begin{bmatrix} k \\ s \end{bmatrix}$ ways of choosing s rows in a product, the generator matrix G_{RM} is of size $\begin{bmatrix} k \\ s \end{bmatrix} \times 2^k$, and thus the dimension of an r^{th} order RM code is equal to $1 + \begin{bmatrix} k \\ 1 \end{bmatrix} + \begin{bmatrix} k \\ 2 \end{bmatrix} + \dots + \begin{bmatrix} k \\ r \end{bmatrix}$.

Now consider the more general class of GRM codes. They are defined over a general

Galois field $GF(q)$ in contrast to the original RM codes which were introduced as binary codes. When $q = 2$, the GRM code reduces to a code which is equivalent to the original RM code, i.e., a code whose code vector components are obtained by a permutation of those of the latter. Our discussions in this section are limited to $q = 2$.

Since binary GRM codes have block length 2^k , an appropriate study of these codes using GSFs would be to consider the GP representation of the individual code vectors rather than a single LGP representation of the whole code (Such a characterization of linear block codes is dealt with in Chapter 5). Further such a description gives more insight into the algebraic structure of these codes.

4.4.1 Representation of the Basis Vectors of Binary GRM Codes by GPs

Let us examine the nature of the GP representation of the basis vectors of a binary GRM code. It is known that the Generator matrix G_{GRM} of a binary GRM code can be obtained if, in the generator matrix of the original RM code G_{RM} , the columns of the matrix G_{RM_1} are chosen such that the binary k -tuples in them are taken in the field order. Let us denote the corresponding matrix in G_{GRM} as G_{GRM_1} . Then G_{GRM_s} , $2 \leq s \leq r$, can be obtained from this G_{GRM_1} in the same manner as in the case of RM codes, i.e., by taking componentwise products of its rows, s at a time.

Now let us consider the matrix G_{GRM_1} . Evidently, the m rows of this matrix represent m LBFs. We have seen that any LBF has a single term LFP representation. The degree of this LFP is 2^{m-1} . Now consider the matrix G_{GRM_2} , the rows of which are formed by taking componentwise products of all possible two such LBFs in the Boolean domain. Let the GP representation of these two LBFs, say $f_1(x)$ and $f_2(x)$ ($f_1(x) \neq f_2(x)$), each of degree 2^{m-1} , be given by

$$f_1(x) = \sum_{i=0}^{m-1} a_i x^{2^i},$$

$$\text{and } f_2(x) = \sum_{j=0}^{m-1} b_j x^{2^j}.$$

When taking componentwise products of these two functions in the Boolean domain is same as multiplying $f_1(x)$ and $f_2(x)$ modulo $(x^{2^m} + x)$. Clearly, the degree of the polynomial product is equal to $2^{m-1} + 2^{m-2}$. Arguing on the same lines, the degree of the polynomial product of s LBFs is equal to $2^{m-1} + 2^{m-2} + \dots + 2^{m-s}$ which may be simplified as $2^{m-s}(2^s - 1) = 2^m - 2^{m-s}$.

Thus the degree of the GP representation of the rows of G_{GRM_s} , $1 \leq s \leq r$, is equal to $2^m - 2^{m-s}$. Alternatively, $2^{m-s} - 1$ consecutive GP coefficients of the basis vectors in G_{GRM_s} , $1 \leq s \leq r$, are equal to zero.

4.4.2 GRM Codes as Ideals in a Monoid Algebra

We use the results given by Blahut [20] on GRM codes to prove that they are in fact ideals in the monoid algebra of BFs.

A GRM code of order r and block length q^k is obtained by appending an overall parity check symbol to a cyclic code of block length q^{k-1} and order r known as a *cyclic GRM code*. Blahut defines a cyclic GRM code as follows:

Definition 4.4.2: A cyclic GRM code of order r and block length q^{k-1} over $\text{GF}(q)$ is a cyclic code whose generator polynomial $g(x)$ has zeroes at all γ^j (γ being a primitive element of $\text{GF}(2^k)$), $j = 1, 2, \dots, q^{k-1}$, such that

$$0 < w(j) \leq (q-1)k-r-1,$$

where $w(j)$ is called the *weight* of the integer $j = j_0 + j_1q + j_2q^2 + \dots + j_{k-1}q^{k-1}$, and is defined as

$$w(j) = j_0 + j_1 + j_2 + \dots + j_{k-1}, \quad (4.4.3)$$

where the addition is ordinary integer addition.

Since j and $jq \bmod (q^k-1)$ have the same weight, if γ^j is a zero of $g(x)$ so are the conjugates of γ^j . Blahut [20] has restated the definition of cyclic GRM codes in terms of the discrete Fourier transform (DFT) or spectrum of the code vectors as follows:

Definition 4.4.3: A cyclic GRM code of order r and block length q^k-1 over $GF(q)$ is the set of code vectors whose spectral component j equals zero for all j satisfying

$$0 < w(j) \leq (q-1)k-r-1.$$

We first reformulate the definitions of GRM codes given by Blahut in terms of GPs in the following theorem (Only binary GRM codes are considered):

Theorem 4.4.1: A binary GRM code of block length 2^k and order r ($r < k$) is the set of code vectors (BFs of length 2^k) whose GP coefficients a_j are zero, for all j satisfying

$$0 \leq w(j) \leq k-r-1.$$

Proof: The Galois transform matrix of order 2^k is obtained by extending a DFT matrix of order 2^k-1 , and the coefficients a_j , $j = 1, \dots, 2^k-2$, are exactly same as the corresponding DFT coefficients. Thus the condition $a_j = 0$, $j = 1, \dots, 2^k-2$, satisfying $0 < w(j) \leq k-r-1$ is valid for the GP coefficients also. In addition, the coefficient $a_0 = 0$ for all $r < k$. This is because this coefficient is obtained by summing modulo 2, all the 2^k components of a code vector, and the number of ones in a code vector of any r^{th} order RM code ($r < k$) is even.

Q.E.D.

From the definition of GRM codes in terms of GP coefficients, we see that a binary GRM code of block length 2^k and order r ($r < k$) may be obtained by forcing to zero all the

coefficients a_j , for $j = 0$, and for all j satisfying the condition $0 < w(j) \leq k-r-1$, and assigning the remaining GP coefficients to assume all possible values from $GF(2^d)$, where d is the order of a Frobenius class (which is the number of GP coefficients in that Frobenius class) as allowed by the conjugacy constraints.

If $a_j = 0$, then a_{jq} is also equal to zero, where jq is taken modulo 2^k-1 , and therefore it suffices to consider only one member a_j of each Frobenius class whose j satisfies the condition $0 < w(j) \leq k-r-1$, since the other members are forced to zero automatically by the conjugacy constraints. On similar lines, it is sufficient to consider one member a_j of each Frobenius class, for $j = -\infty$ and for all j 's which satisfy the condition $w(j) > k-r-1$, which are allowed to assume all possible values from $GF(2^d)$, d being the order of the Frobenius class. This situation is similar to the formation of ideals in the monoid algebra of BFs which was discussed in Section 4.2.1, where any ideal in this algebra was expressed as a direct sum of certain minimal ideals and where these minimal ideals were obtained by assigning all the values of $GF(2^d)$ to one Frobenius class at a time and forcing the elements of the remaining Frobenius classes to zero.

In the wake of the above discussion, we may form an r^{th} order GRM code of block length 2^k as the direct sum of minimal ideals in the monoid algebra of BFs as follows:

Choose one member j of the conjugacy class modulo 2^k-1 which satisfies the condition $w(j) > k-r-1$. Now allow the GP coefficient a_j with this j as index to assume all possible values from $GF(2^d)$, where d is the order of the Frobenius class. This fixes the remaining coefficients in the Frobenius class containing a_j since they are related to a_j by conjugacy constraints. Further, these members also satisfy the condition $w(j) > k-r-1$, since j and $jq \bmod (2^k-1)$ have the same weight. In this manner, we have formed a minimal ideal. Similarly form minimal ideals corresponding to $j = -\infty$ and all j which satisfies the condition on weight, i.e., $w(j) > k-r-1$. Now the r^{th} order GRM code is a direct sum of these minimal ideals.

Examples

Example 4.4.1: Consider a first order GRM code of block length 2^3 . The GP coefficients in this case have indices $-\infty, 0, 1, \dots, 6$. These are grouped into Frobenius classes as $\{-\infty\}$, $\{0\}$, $\{1, 2, 4\}$ and $\{3, 6, 5\}$. We may see that all the elements j in a Frobenius class have the same weight. In this case, the weights are respectively 1 and 2 (for $j > 0$). We choose the index $j = -\infty$, and those indices j which satisfy $w(j) > 3-1-1$, i.e., $w(j) > 1$. This is satisfied by the class $\{3, 6, 5\}$. Since the order of this class is 3, we allow one of the GP coefficients with these as indices, to assume all possible values from $GF(2^3)$ leaving the remaining coefficients to be fixed by the conjugacy constraints. Thus let us choose a_3 to assume all possible values (8 in number) from $GF(2^3)$. This gives us one minimal ideal containing 8 elements. Now, the order of the class $\{-\infty\}$ is 1, and hence $a_{-\infty}$ assumes values from $GF(2)$, giving another minimal ideal of order 2. Taking the direct sum of these two minimal ideals gives an ideal containing 16 elements whose inverse Galois transform give the code vectors of a binary first order GRM code of block length 8 and dimension 4. The GP coefficients of the code vectors of this code are listed in Table 4.2 as a power of a primitive element γ in $GF(2^3)$. The coefficients are listed in the order a_j , $j = -\infty, 0, 1, \dots, 6$. Only the exponents of γ are listed.

Example 4.4.2: Consider a second order GRM code of block length 2^4 . The GP coefficients in this case have indices $-\infty, 0, 1, \dots, 14$. These are grouped into 6 Frobenius classes as $\{-\infty\}$, $\{0\}$, $\{1, 2, 4, 8\}$, $\{3, 6, 12, 9\}$, $\{5, 10\}$, and $\{7, 14, 13, 11\}$. In this case, the weights are respectively 1, 2, 2 and 3 (for $j > 0$). We choose the index $j = -\infty$, and those indices j which satisfy $w(j) > 4-2-1$, i.e., $w(j) > 1$. This is satisfied by the classes containing 3, 5 and 7. The order of these classes are respectively 4, 2 and 4. Thus we get two minimal ideals of order 16 corresponding to the coefficients a_3 and a_7 which are allowed to assume all the 16 values from $GF(2^4)$, one minimal ideal of order 4 corresponding to a_5 which is allowed to take values from $GF(2^2)$, and a fourth minimal ideal of order 2 corresponding to

which assumes values from $GF(2)$. Taking the direct sum of these four minimal ideals we have an ideal containing $2^4 \times 2^4 \times 2^2 \times 2 = 2^{11}$ elements whose inverse Galois transforms give the code vectors of a binary second order GRM code of block length 16 and dimension 11. This code, being too large, is not listed.

Table 4.2: GP Coefficients of the Code Vectors of the First Order GRM Code of Block Length 8 Considered in Example 4.4.1

$a_{-\infty}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	0	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	1	$-\infty$	4	2
$-\infty$	$-\infty$	$-\infty$	$-\infty$	2	$-\infty$	1	4
$-\infty$	$-\infty$	$-\infty$	$-\infty$	3	$-\infty$	5	6
$-\infty$	$-\infty$	$-\infty$	$-\infty$	4	$-\infty$	2	1
$-\infty$	$-\infty$	$-\infty$	$-\infty$	5	$-\infty$	6	3
$-\infty$	$-\infty$	$-\infty$	$-\infty$	6	$-\infty$	3	5
0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	0	0
0	$-\infty$	$-\infty$	$-\infty$	1	$-\infty$	4	2
0	$-\infty$	$-\infty$	$-\infty$	2	$-\infty$	1	4
0	$-\infty$	$-\infty$	$-\infty$	3	$-\infty$	5	6
0	$-\infty$	$-\infty$	$-\infty$	4	$-\infty$	2	1
0	$-\infty$	$-\infty$	$-\infty$	5	$-\infty$	6	3
0	$-\infty$	$-\infty$	$-\infty$	6	$-\infty$	3	5

In the next section, we consider the classification problems of BFs.

4.5 Classification of Boolean Functions

An area in which spectral techniques were successfully applied was in the classification of BFs. The need for classification arises due to the fact that the number of possible different k -variable BFs is considerably large even for small values of k , this number being 2^{2^k} , thus making the task of enumeration of these functions difficult. So, it

ould be desirable, if they are classified into equivalence classes under some equivalence relations. We say that two functions are equivalent under some equivalence relation, if one can be transformed into the other by that relation. The larger the number of such equivalence relations which can be defined on these functions, the lesser the number of classes would be. A canonic function may be chosen as a representative member for each class, and the remaining functions in that class may be generated from the circuit realization of the former, by implementing additionally the appropriate operations corresponding to the equivalence relations used for the classification purpose. Another advantage which results from classification of functions is that testing and fault diagnosis procedures may be standardized for each entry in a class.

A number of equivalence relations have been proposed in literature to reduce the number of classes for a given number of inputs. Further, their effect on the transform coefficients were also studied, as a consequence of which, given the transform coefficients of any arbitrary function in any class, the operations corresponding to the equivalence relations may be carried out in the spectral domain to obtain the canonic function of that class. Five of these equivalence relations, commonly known as the *five invariance operations*, are described in the following subsection:

5.1 The Five Invariance Operations

The five invariance operations act on the domain and range of BFs. These five operations were called *invariance operations* in connection with the classification of functions based on these operations using Rademacher–Walsh (R–W) functions because they did not change the magnitude of the R–W coefficients.

5.1.1 Invariance Operations on the Domain of Boolean Functions

There are three operations which operate on the input variables of a BF. They are described as follows:

1) Complementation of Input Variables

Two k -variable BFs f and g are said to be equivalent under the above operation if one can be obtained from the other by complementing one or more of its input variables. For example, let $k = 2$ and let $f = \bar{x}_1 \cup x_2$ and let $g = x_1 \cup x_2$, where x_1 and x_2 are the input variables and \cup denotes inclusive OR operation. Then f and g are equivalent under the complementation operation because one can be obtained from the other by complementing x_1 or \bar{x}_1 , as the case may be. Thus any function in this equivalence class may be realized by the same circuit except that some of the inputs are replaced by their complements.

2) Permutation of Input Variables

Two k -variable BFs f and g are said to be equivalent under the above operation if one can be obtained from the other by a permutation of one or more of its input variables. For example, consider two 3-variable BFs f and g . Let $f = x_1 \cup \bar{x}_2 \cdot x_3$ and let $g = x_2 \cup \bar{x}_3 \cdot x_1$, where x_1 and x_2 are the input variables. Then f and g are equivalent under the permutation operation since $g(x_1, x_2, x_3) = f(x_2, x_3, x_1)$.

(3) Ex-oring the Input Variables

Two k -variable BFs f and g are said to be equivalent under the above operation if one can be obtained from the other by replacing one or more of its inputs by the modulo-2 sum (ex-or) of some of the inputs. For example, if f and g are 3-variable BFs and $f = x_1 + \bar{x}_2 \cdot x_3$, and $g = x_1 + x_2 \cup \bar{x}_2 \cdot x_3$, where $+$ denotes modulo-2 sum, then f and g are

equivalent, as g is obtained from f by replacing the input x_1 by the ex-or of x_1 and x_2 .

4.5.1.2 Invariance Operations on the Range of Boolean Functions

There are two relations which operate on the range of a BF. They are described as follows:

(4) Complementation of the Output

Two k -variable BFs f and g are said to be equivalent under the above operation if one can be obtained from the other by complementing its output. Thus g and f are equivalent if $g = \bar{f} = f + 1$, or $f = \bar{g} = g + 1$.

(5) Ex-oring the Input Variables with the Output

Two k -variable BFs f and g are said to be equivalent under the above operation if one can be obtained from the other by ex-oring one or more of its inputs with its output.

Thus f and g are equivalent if $g = f + \sum_{i=1}^k \ell_i x_i$, where $\ell_i \in \{0,1\}$. In other words, f and g are equivalent if g is obtained from f by a modulo-2 sum of the latter output with a LBF obtained from its inputs.

4.5.1.3 Combining the Operations

It is always desirable to have an equivalence relation which allows a combination of any of the above operations since this leads to a reduced number of classes. For example, if we combine (2) and (3), we get the *General Linear Group*, say $GL_k(Z_2)$, which is the group of all invertible linear transformations acting on a k -dimensional vector space over the field Z_2 .

We say that two k -variable BFs f and g are equivalent under $GL_k(Z_2)$, if there is a

non-singular matrix \underline{A} such that $g(x_1, x_2, \dots, x_k) = f((x_1, x_2, \dots, x_k) \cdot \underline{A})$. This leads to much smaller number of classes as the order of $GL_k(Z_2)$ is considerably large.

We may still combine (1) with $GL_k(Z_2)$ so as to have a larger group which allows linear transformations as well as complementation of input variables. This group is called *the Affine Group* and is denoted as $A_k(Z_2)$. Thus two k -variable BFs f and g are equivalent under $A_k(Z_2)$ if there is a non-singular matrix A and a constant $\beta \in GF(2^k)$ such that

$$g(x_1, x_2, \dots, x_k) = f((x_1, x_2, \dots, x_k) \cdot A + \beta). \quad (4.5.1)$$

Thus the affine equivalence relation comprises of all the three domain operations.

Similarly, we may combine the two range operations with the input operations by adding an affine polynomial $c + \sum_{i=1}^k \ell_i x_i$ to the output of a logic circuit f whose input was defined by an affine transformation of $x = x_1, x_2, \dots, x_k$, where c and $a_i \in \{0,1\}$ (The resulting group was called *Restricted Affine Group (RAG)* by Lechner [16]). Thus two k -variable BFs f and g are equivalent under the five invariance operations if g is obtained from f such that

$$g(x_1, x_2, \dots, x_k) = f((x_1, x_2, \dots, x_k) \cdot A + \beta) + \sum_{i=1}^k \ell_i x_i + c, \quad (4.5.2)$$

where A is a non-singular matrix of size $k \times k$, $\beta \in GF(2^k)$, and $c, \ell_i \in GF(2)$.

Classification of BFs based on these five operations were carried out using Rademacher-Walsh coefficients by researchers and the number of classes for $k = 2, 3, 4$ and 5 were respectively found to be 2, 3, 8 and 48.

4.5.2 Effect of the Five Invariance Operations on the GP Coefficients

In this subsection, we study the effect of the five invariance operations on the GP coefficients of BFs. In all the cases, the GP coefficients of the original k -variable BF, $f(x)$,

will be denoted by a_1 , and that of the new function obtained by the respective operation will be denoted by \hat{a}_1 , where $i = -\infty, 0, 1, \dots, 2^k-2$.

(1) Complementation of Input Variables

Complementation of the input variable x of a k -variable BF, $f(x)$, is same as adding a constant, say $\beta \in GF(2^k)$, to x . Then the relation between the GP coefficients of the original function and that of the new function may be derived as follows:

The coefficient $\hat{a}_{-\infty} = f(\gamma^{-\infty} + \beta) = f(\beta),$ (4.5.3a)

where $f(\beta) = \sum_1 a_1 \beta^{-i}, i = -\infty, 0, 1, \dots, 2^k-2.$

The coefficients $\hat{a}_1, i = 0, 1, \dots, 2^k-2$, are given by

$$\begin{aligned} \hat{a}_1 &= \sum_x x^i f(x + \beta). \\ &= \sum (x + \beta)^i f(x). \\ &= \sum_x \sum_{t=0}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 x^{i-t} \beta^t f(x). \\ &= \sum_{t=0}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t \sum_x x^{i-t} f(x). \end{aligned}$$

$$= \sum_{t=0}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t a_{1-t}, i = 0, 1, \dots, 2^k-2, \quad (4.5.3b)$$

where $\left[\begin{matrix} i \\ t \end{matrix} \right]_2$ is the binomial coefficient modulo 2.

If we put (4.5.3) in matrix form (with $\hat{a}_{-\infty}$ also expressed in terms of the a_1 's), as

$$\hat{\underline{a}} = \underline{R} \underline{a}, \quad (4.5.4)$$

where $\underline{a} = [a_0, a_1, a_2, \dots, a_\xi, a_{-\omega}]^T$
 and $\hat{\underline{a}} = [\hat{a}_0, \hat{a}_1, \hat{a}_2, \dots, \hat{a}_\xi, \hat{a}_{-\omega}]^T$

where $\xi = 2^k - 2$ then it may be seen that \underline{R} may be expressed as a direct product of k core matrices \underline{R}_i , $i = 0, 1, \dots, k-1$, each of size 2×2 , as

$$\underline{R} = \underline{R}_{k-1} \otimes \underline{R}_{k-2} \otimes \dots \otimes \underline{R}_1 \otimes \underline{R}_0, \quad (4.5.5)$$

where $\underline{R}_i = \begin{bmatrix} 1 & 0 \\ \beta^{2^i} & 1 \end{bmatrix}$

and 2^i is taken modulo $2^k - 1$.

Thus for example, if $k = 4$, then

$$\underline{R} = \begin{bmatrix} 1 & 0 \\ \beta^8 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \beta^4 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \beta^2 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ \beta & 1 \end{bmatrix}$$

2) Linear Transformation of Input Variables

We combine the two operations of permutation and ex-oring of input variables into the single operation of linear transformation of the same. Let \underline{A} be a non-singular matrix of size $k \times k$. Then we perform linear transformation on x by multiplying \underline{A} by the k -tuple x . Then the relation between the coefficients a_i and \hat{a}_i may be derived as follows:

The coefficient $\hat{a}_{-\omega} = a_{-\omega}, \quad (4.5.6a)$

since $\hat{a}_{-\omega} = f((00\dots 0)\underline{A}) = f(00\dots 0) = f(\gamma^{-\omega}) = a_{-\omega}.$

The coefficients \hat{a}_i , $i = 0, 1, \dots, 2^k - 2$, are obtained by replacing x by $x.\underline{A}$.

Thus we have

$$\begin{aligned} \hat{a}_i &= \sum_x x^i f(x.\underline{A}). \\ &= \sum_x (x.\underline{A}^{-1})^i f(x). \end{aligned}$$

Now the mapping from x to $x.\underline{A}^{-1}$ is a linear mapping and hence it may be represented by a LP, say $F_A(x)$. Let $F_A(x)$ be given by

$$F(x) = \sum_{j=0}^{k-1} d_j x^{2^j}. \quad (4.5.6b)$$

Thus

$$\hat{a}_i = \sum_x \left(\sum_{j=0}^{k-1} d_j x^{2^j} \right)^i f(x). \quad (4.5.6c)$$

For those coefficients \hat{a}_i , whose i is of the form 2^m , $m = 0, 1, \dots, k-1$, (4.5.6c) may be further simplified as follows:

$$\hat{a}_{2^m} = \sum_x \left(\sum_{j=0}^{k-1} d_j x^{2^j} \right)^{2^m} f(x)$$

$$= \sum_x \left(\sum_{j=0}^{k-1} d_j^{2^m} x^{2^{j+m}} \right) f(x)$$

$$= \sum_{j=0}^{k-1} d_j^{2^m} \sum_x x^{2^{j+m}} f(x)$$

$$= \sum_{j=0}^{k-1} d_j^{2^m} a_{2^{j+m}}, \quad m = 0, 1, \dots, k-1. \quad (4.5.6d)$$

3) Complementation of the Output

Complementing the output of a BF may be accomplished by adding a '1' to it. The relation between the coefficients a_i and \hat{a}_i may be obtained as follows:

The coefficient

$$\hat{a}_{-\infty} = f(\gamma^{-\infty}) + 1 = \bar{f}(\gamma^{-\infty}) = \bar{a}_{-\infty}, \quad (4.5.7a)$$

where the over head bar denotes complementation.

The coefficients \hat{a}_i , $i = 0, 1, \dots, 2^k - 2$, are obtained as

$$\hat{a}_i = \sum_x x^i (f(x) + 1) = \sum_x x^i f(x) = a_i. \quad (4.5.7b)$$

Thus the output complementation of $f(x)$ leaves the coefficients invariant except the constant term, which gets complemented.

(4) Ex-oring the Input Variables with the Output

Ex-oring the input variables x_i with the output of a BF is same as adding a LBF to it. The relation between the coefficients a_i and \hat{a}_i may be obtained as follows:

Let the LBF be given by $\sum_{i=1}^k \ell_i x_i$, where ℓ_i 's $\in \{0,1\}$. Because, the function is linear, and the mapping is from $GF(2^k)$ to $GF(2)$, it has a single term LFP representation (with the only nonzero coefficients corresponding to those of x^{2^i} , $i = 0, 1, \dots, k-1$).

Thus when a LBF of the form above is added to the output of the BF, its effect on the coefficients of the same would be to modify only those coefficients of the BF corresponding to x^{2^i} , $i = 0, 1, \dots, k-1$, keeping the remaining coefficients intact.

Let $f(x) = a_{-\infty} + \sum_j \text{frs}(a_j x^{-j})$ denote the k -variable BF (where j is a member of the conjugacy class modulo 2^k-1), and let $\ell(x) = \text{frs}(\gamma x)$ denote the LBF added to $f(x)$, where γ and $a_j \in GF(2^k)$.

Then the resulting function, say $\hat{f}(x)$, is given by

$$\hat{f}(x) = a_{-\infty} + \sum_{j \neq \xi} \text{frs}(a_j x^{-j}) + \text{frs}((a_{\xi} + \gamma)x), \quad (4.5.8)$$

where $\xi = 2^k-2$.

4.5.3 Class Identification by Verifying the GP Coefficients

Given a BF, our first task would be to identify the class to which it belongs so that it may be synthesized from the representative member of that class. This identification is possible for 2 and 3 variable BFs by checking their GP coefficients, one in the case of 2-variable BFs, and two in the case of 3-variable BFs. As mentioned earlier, all the 2-variable BFs are classified into 2 classes and all the 3-variable BFs are classified into 3 classes in the classification procedure using the five invariance operations. In 2-variable case, all the 16 BFs are put into two classes each having 8 members, and in the 3-variable case, all the 256 BFs are put into 3 classes, each of strength 16, 128 and 112 respectively. A

study of the nature of the GP coefficients of the functions of these classes reveals the following:

(1) $k = 2$:

Class I No. of functions = 8.

- (a) $a_{-\infty}$ assumes either 0 or 1.
- (b) a_0 assumes 0 only.
- (c) $\{a_1, a_2\}$ assumes all the 4 values from $GF(2^2)$.

Class II No. of functions = 8.

- (a) $a_{-\infty}$ assumes either 0 or 1.
- (b) a_0 assumes 1 only.
- (c) $\{a_1, a_2\}$ assumes all the 4 values from $GF(2^2)$.

(2) $k = 3$:

Class I No. of functions = 16.

- (a) $a_{-\infty}$ assumes either 0 or 1.
- (b) a_0 assumes 0 only.
- (c) $\{a_1, a_2, a_4\}$ assumes 0 only.
- (d) $\{a_3, a_6, a_5\}$ assumes all the 8 values from $GF(2^3)$.

Class II No. of functions = 128.

- (a) $a_{-\infty}$ assumes either 0 or 1.
- (b) a_0 assumes 1 only.
- (c) $\{a_1, a_2, a_4\}$ assumes all the 8 values from $GF(2^3)$.
- (d) $\{a_3, a_6, a_5\}$ assumes all the 8 values from $GF(2^3)$.

Class III

No. of functions = 112.

- (a) $a_{-\infty}$ assumes either 0 or 1.
- (b) a_0 assumes 0 only.
- (c) $\{a_1, a_2, a_4\}$ assumes all the 7 nonzero values from $GF(2^3)$.
- (d) $\{a_3, a_6, a_5\}$ assumes all the 8 values from $GF(2^3)$.

Based on the above study we formulate a simple class identification procedure for 2 and 3 variable BFs by verification of the GP coefficients as follows:

(1) $k = 2$

The only coefficient which needs to be computed is a_0 , since we saw that the GP coefficients of the functions in the first class have $a_0 = 0$ and those in the second class have $a_0 = 1$. Thus the identification procedure is

Calculate a_0 .

If it is 0, then the function belongs to the first class, and if 1, it belongs to the second class.

(2) $k = 3$

Here the steps involved are as follows:

(1) Calculate a_0 ; If 1, then the function belongs to the second class.

If 0, then calculate a_1 ; If $a_1 = 0$, then the function belongs to the first class, and if nonzero, it belongs to the third class.

4.5.4 Operations Based on the Monoid Algebra Structure of Boolean Functions

In this section, we consider some operations based on the monoid algebra structure of k -variable BFs.

4.5.4.1 Convolution Operation with a Function whose GP Coefficients are $a_{-\infty} = 0, a_i = \gamma^{-i}, i = 0, 1, \dots, 2^k-2$.

First we consider a particular case of the convolution operation defined in Chapter 2, where one of the functions involved is taken as one having GP coefficients $a_i, i = -\infty, 0, 1, \dots, 2^k-2$, respectively as $0, 1, \gamma^{2^k-2}, \gamma^{2^k-3}, \dots, \gamma^2, \gamma$, where γ is a primitive element of $GF(2^k)$. Let us denote this function as f_s . The convolution of any function with f_s is performed by pointwise multiplication of the GP coefficients of that function with that of f_s . In the Boolean domain, this has the effect of keeping the first function value $f(\gamma^{-\infty})$ fixed and cyclically shifting the remaining values one position towards left. We group the BFs based on this operation. Since the coefficients $a_{-\infty}$ and a_0 can assume only 0 or 1, we need consider only one fourth of the total BFs, corresponding to, say, both these coefficients being 0. The remaining groups may then be obtained by just changing these coefficients to $\{0, 1\}, \{1, 0\}$, and $\{1, 1\}$. Thus the actual number of classes will be clearly equal to 4 times the number of classes obtained for one-fourth of the functions.

(1) $k = 2$

The number of Frobenius classes in this case is 3, out of which we fix the first two as zero. Thus we take four of the sixteen functions, i.e., those functions whose GP coefficients $a_{-\infty}$ and a_0 are zero, and group them by multiplying their GP coefficients pointwise with those of f_s . Here f_s is a function whose GP coefficients $a_i, i = -\infty, 0, 1, 2$, are $0, 1, \gamma^2, \gamma$ respectively. It may be seen that 2 classes may be formed out of the four functions. They are given in Table 4.3a, with the GP coefficients in the order $a_{-\infty}, a_0, a_1, a_2$ listed as a power of a primitive element γ in $GF(2^k)$. Only the exponents of γ are listed.

A total of 8 classes may be formed out of a total of 16 BFs by putting $\{a_{-\infty}, a_0\}$ as $\{0, 0\}, \{0, 1\}, \{1, 0\}$ and $\{1, 1\}$.

(2) $k = 3$

The number of Frobenius classes in this case is 4. As before, we consider one fourth (64) of the 256 functions corresponding to those whose GP coefficients $a_{-\infty}$ and a_0 are zero. In this case, the GP coefficients $a_i, i = -\infty, 0, 1, \dots, 6$, of f_g are $0, 1, \gamma^6, \gamma^5, \gamma^4, \gamma^3, \gamma^2, \gamma$ respectively. It may be seen that 10 classes may be formed out of the 64 functions. They are listed in Table 4.3b with the GP coefficients in the order $a_{-\infty}, a_0, a_1, a_2, \dots, a_6$ listed as a power of γ . Only the exponents of γ are listed.

Table 4.3 Classification of k -Variable Boolean Functions using Convolution
as Defined in Monoid Algebra with a Function whose GP Coefficients are

$$a_{-\infty} = 0, a_i = \gamma^{-i}, i = 0, 1, \dots, 2^k - 2.$$

(a) $k = 2$

	$a_{-\infty}$	a_0	a_1	a_2
C1	$-\infty$	$-\infty$	$-\infty$	$-\infty$
C2	$-\infty$	$-\infty$	0	0
	$-\infty$	$-\infty$	2	1
	$-\infty$	$-\infty$	1	2

(b) $k = 3$

	$a_{-\infty}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
C1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
C2	$-\infty$	$-\infty$	0	0	$-\infty$	0	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	6	5	$-\infty$	3	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	5	3	$-\infty$	6	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	4	1	$-\infty$	2	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	3	6	$-\infty$	5	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	2	4	$-\infty$	1	$-\infty$	$-\infty$
	$-\infty$	$-\infty$	1	2	$-\infty$	4	$-\infty$	$-\infty$

Table 4.3b (continued)

	$a_{-\infty}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
C3	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	0	0
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	4	$-\infty$	2	1
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	1	$-\infty$	4	2
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	5	$-\infty$	6	3
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	2	$-\infty$	1	4
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	6	$-\infty$	3	5
	$-\infty$	$-\infty$	$-\infty$	$-\infty$	3	$-\infty$	5	6
C4	$-\infty$	$-\infty$	2	4	0	1	0	0
	$-\infty$	$-\infty$	1	2	4	4	2	1
	$-\infty$	$-\infty$	0	0	1	0	4	2
	$-\infty$	$-\infty$	6	5	5	3	6	3
	$-\infty$	$-\infty$	5	3	2	6	1	4
	$-\infty$	$-\infty$	4	1	6	2	3	5
	$-\infty$	$-\infty$	3	6	3	5	5	6
C5	$-\infty$	$-\infty$	4	1	0	2	0	0
	$-\infty$	$-\infty$	3	6	4	5	2	1
	$-\infty$	$-\infty$	2	4	1	1	4	2
	$-\infty$	$-\infty$	1	2	5	4	6	3
	$-\infty$	$-\infty$	0	0	2	0	1	4
	$-\infty$	$-\infty$	6	5	6	3	3	5
	$-\infty$	$-\infty$	5	3	3	6	5	6
C6	$-\infty$	$-\infty$	6	5	0	3	0	0
	$-\infty$	$-\infty$	5	3	4	6	2	1
	$-\infty$	$-\infty$	4	1	1	2	4	2
	$-\infty$	$-\infty$	3	6	5	5	6	3
	$-\infty$	$-\infty$	2	4	2	1	1	4
	$-\infty$	$-\infty$	1	2	6	4	3	5
	$-\infty$	$-\infty$	0	0	3	0	5	6
C7	$-\infty$	$-\infty$	1	2	0	4	0	0
	$-\infty$	$-\infty$	0	0	4	0	2	1
	$-\infty$	$-\infty$	6	5	1	3	4	2
	$-\infty$	$-\infty$	5	3	5	6	6	3
	$-\infty$	$-\infty$	4	1	2	2	1	4
	$-\infty$	$-\infty$	3	6	6	5	3	5
	$-\infty$	$-\infty$	2	4	3	1	5	6
C8	$-\infty$	$-\infty$	3	6	0	5	0	0
	$-\infty$	$-\infty$	2	4	4	1	2	1
	$-\infty$	$-\infty$	1	2	1	4	4	2
	$-\infty$	$-\infty$	0	0	5	0	6	3
	$-\infty$	$-\infty$	6	5	2	3	1	4
	$-\infty$	$-\infty$	5	3	6	6	3	5
	$-\infty$	$-\infty$	4	1	3	2	5	6

Table 4.3b (continued)

	$a_{-\infty}$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
C9	$-\infty$	$-\infty$	5	3	0	6	0	0
	$-\infty$	$-\infty$	4	1	4	2	2	1
	$-\infty$	$-\infty$	3	6	1	5	4	2
	$-\infty$	$-\infty$	2	4	5	1	6	3
	$-\infty$	$-\infty$	1	2	2	4	1	4
	$-\infty$	$-\infty$	0	0	6	0	3	5
	$-\infty$	$-\infty$	6	5	3	3	5	6
C10	$-\infty$	$-\infty$	0	0	0	0	0	0
	$-\infty$	$-\infty$	6	5	4	3	2	1
	$-\infty$	$-\infty$	5	3	1	6	4	2
	$-\infty$	$-\infty$	4	1	5	2	6	3
	$-\infty$	$-\infty$	3	6	2	5	1	4
	$-\infty$	$-\infty$	2	4	6	1	3	5
	$-\infty$	$-\infty$	1	2	3	4	5	6

4.5.4.2 Convolution Operation on Arbitrary Boolean Functions

We can further reduce the number of classes in the 3-variable case by using the convolution operation on arbitrary functions. Classes C2 and C3 of Table 4.2b each have the structure of a group under the convolution operation, and thus arbitrary functions from within these classes, which when convolved gives functions only within the respective classes.

Now consider class C4. Convolution of any two functions of this class gives a function from class C5. Thus classes C4 and C5 may be now combined. Similarly, convolution of any function from C4 with one from C5 gives a function from class C6. We may further convolve functions in C4 with functions in C7, C8 and C9 to get functions from C8, C9 and C10 respectively. Therefore classes from C4 to C10 may be combined in this manner by convolution on arbitrary functions. In this process we have constructed three nontrivial groups which are closed under convolution (the class containing the

singleton 0 function being the fourth group). We rename the resulting classes as Class I, II, III and IV respectively. These classes along with the class members and the number of members in each class are listed in Table 4.4.

Table 4.4 Classification of 3-Variable Boolean Functions using Convolution as Defined in Monoid Algebra on Arbitrary Functions

Class	Class Members	No. of members
I	C1	1
II	C2	7
III	C3	7
IV	C_i $i = 4, \dots, 10$	49

In total, 16 groups may be formed out of 256 functions. Thus we have reduced the number of classes from 40 to 16 in the 3-variable case. Incidentally, these 16 classes correspond to the 16 ideals in this algebra, each class assuming elements from each of these ideals. For example, class II and class III contain the nonzero elements of I_3 and I_4 respectively, and class IV has elements from the ideal $I_3 \oplus I_4$ where \oplus denotes direct sum. *The fact that ideals in this algebra may be generated by a direct sum of minimal ideals, also gives a method to the generation of BFs, by summing elements from the minimal ideals. This is the Frobenius sum model which is discussed in the next section.*

4.6 Finite Field Models for Boolean Function Synthesis

In this section, we propose two models for BF synthesis, one based on the five operations and the other based on Frobenius sum computation.

4.6.1 Model Based on the Five Invariance Operations

This model is based on the study conducted in Section 4.5.2, based on the five invariance operations. We know that two k -variable BFs f and g are equivalent under the five invariance operations if g is obtained from f such that they satisfy (4.5.2).

The synthesis procedure for the function g would be as follows:

- (1) Identify the class to which g belongs.
- (2) Take the representative function f for that class.
- (3) Select the suitable domain and range transformations which when applied to f , would synthesize g .

Circuits which implement g using the above model have been proposed in literature. However, we propose a finite field model for realization of g . This model is based on our study of the five operations on the GP coefficients of BFs.

First, let us look at the domain transformation. Here, we perform linear transformation and complementation of input variables. This is done by an affine transformation on the input variable $x = x_1 x_2 \dots x_k$. This affine transformation has an affine polynomial representation say $F_a(x)$, which is the sum of a LP and a constant β belonging to $GF(2^k)$. ie., $F_a(x)$ is given by

$$F_a(x) = \beta + F_\ell^{-1}(x), \quad (4.6.1)$$

where $F_\ell^{-1}(x)$ is a LP representing \underline{A} , which satisfies

$$F_\ell^{-1}(F_\ell(x)) = F_\ell(F_\ell^{-1}(x)) = x, \quad (4.6.2)$$

where $F_\ell(x)$ is given by (4.5.6b).

Thus the affine transformation is performed by the operation of composition $f(F_a(x))$.

Next let us consider the range transformation. This consists of adding a LBF and a binary constant to the output of $f(x)$. The linear function has a single term LFP representation, say $\text{frs}(\gamma x)$, where $\gamma \in GF(2^k)$. The binary constant may be realized from the Frobenius sum of an element γ_c belonging to $GF(2^k)$, say $\text{frs}(\gamma_c)$. This is because

Frobenius sum always gives an element from the ground field. γ_c is chosen such that its Frobenius sum gives the required binary constant. Thus the range transformation is performed by adding the terms $\text{frs}(\gamma x)$ and $\text{frs}(\gamma_c)$ to the GP representation of $f(F_{\mathbf{a}}(x))$. Since $\text{frs}(\cdot)$ is a linear function, we may write

$$\text{frs}(\gamma x) + \text{frs}(\gamma_c) = \text{frs}(\gamma x + \gamma_c). \quad (4.6.3)$$

Thus the finite field model which synthesizes the BF g in any class based on the five invariance operations may be implemented by realizing the equation

$$g(x) = f(F_{\mathbf{a}}(x)) + \text{frs}(\gamma x + \gamma_c) \quad (4.6.4)$$

This model is given in Figure 4.1a.

4.6.2 Model Based on Frobenius Sum Computation

This model is derived from the nature of the GP representation of a BF. We saw that any BF may be represented as a FF, which is a sum of elements of minimal ideals in the corresponding monoid algebra, the number of Frobenius terms in the function given by nfrob of (2.4.3). Thus a BF may be realized by a Frobenius sum model as given in Figure 4.1b.

In Figure 4.1b, we assume that there are m Frobenius terms in the GP representation of the BF under consideration. The first column of blocks numbered 1, 2, ..., m , exponentiate the input variable x . The i^{th} block exponentiates x to the power $-j_i$, (where j_i , $i = 2, \dots, m$, and $-j_i$ taken modulo $2^k - 1$) is a representative member of the Frobenius class modulo $2^k - 1$. j_1 is taken as ∞ , to account for the constant term, in which case, the exponentiated output is taken as unity. The outputs of the exponentiation blocks are multiplied with the corresponding coefficients a_{j_i} , $i = 1, 2, \dots, m$, and then fed to the Frobenius sum computers which compute the Frobenius sums. The outputs of these Frobenius sum computers are fed to an adder block which sums them and gives the

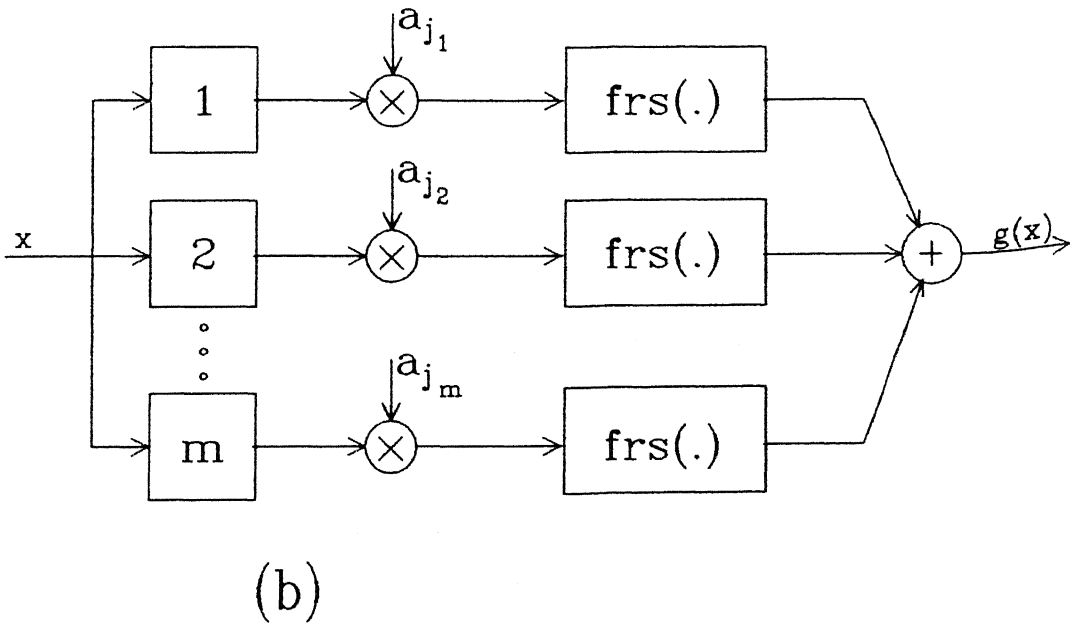
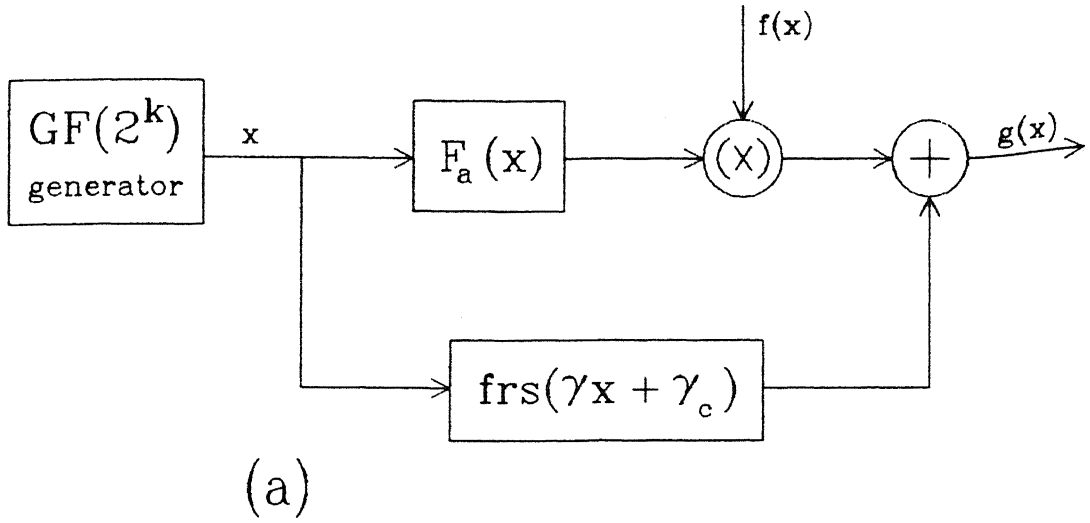


Fig 4.1: Finite Field Models for Boolean Function Synthesis

(a): Model Based on the Five Invariance Operations

(b): Model Based on Frobenius Sum Computation

required BF, $g(x)$.

We conclude this chapter with the next section after a characterization of k -variable β -self dual/anti self dual BFs by GPs.

4.7 Characterization of β -Self Dual / Anti Self Dual Boolean Functions by GPs

In this section, we characterize classes of k -variable β -self dual (SD) and anti self dual (ASD) BFs using GPs.

Definitions

Definition 4.7.1: A BF of k -variables, say $f(x)$, is said to be β -self dual or *partially self dual*, if $\tilde{f}(\tilde{x}) = f(x)$, where $\tilde{x} = x + \beta$, and β is an element of $GF(2^k)$ added to x to get some of the input variables complemented.

As before, \tilde{f} indicates that the function output is complemented. If $\beta = 111\dots 1$ (all ones), we get a completely SD function, which will be called simply as a *SD* function.

Definition 4.7.2: A BF of k -variables, say $f(x)$, is said to be *self dual* if $\bar{f}(\bar{x}) = f(x)$, where \bar{x} indicates that all the input variables $x_1 x_2 \dots x_k$ are individually complemented, and \bar{f} indicates that the function output is complemented.

Definition 4.7.3: A BF of k -variables, say $f(x)$, is said to be β -anti self dual or *partially anti self dual*, if $\tilde{f}(\tilde{x}) = f(x)$, where $\tilde{x} = x + \beta$, and β is an element of $GF(2^k)$ added to x to get some of the input variables complemented.

If $\beta = 111\dots 1$ (all ones), we get a completely ASD function, which will be called simply as an *ASD* function.

Definition 4.7.4: A BF of k -variables, say $f(x)$, is said to be *anti self dual* if $\bar{f}(\bar{x}) = f(x)$,

where \bar{x} indicates that all the input variables $x_1 x_2 \dots x_k$ are individually complemented.

In the next few subsections, we discuss the characterization of both partial and complete SD/ASD BFs using their GPs. We henceforth call these functions as β -SD/ASD functions. When $\beta = 11\dots 1$ (all ones), they become (completely) SD/ASD functions. We derive the constraints on the GP coefficients of these k -variable functions for $k = 2, 3$ and 4.

4.7.1 Derivation Strategy

To begin with, first we recall our discussion on the effect of complementation of input variables of a BF on its GP coefficients. We saw that if we denote the GP coefficients of the original function by a_i , and the GP coefficients of the new function $f(\bar{x})$ resulting from input complementation by \hat{a}_i , $i = -\infty, 0, 1, \dots, 2^k-2$, then the coefficients $\hat{a}_{-\infty}$, and the coefficients \hat{a}_i , $i = 0, 1, \dots, 2^k-2$, are given by (4.5.3).

Secondly, we recall that complementing the output of a function merely complements the constant term $a_{-\infty}$, keeping the rest of the coefficients invariant.

Keeping the above two facts in mind, we derive the constraints on the GP coefficients of a BF to be β -SD/ASD in the following subsections. We limit our discussions to $k = 2, 3$ and 4.

We use the following derivation strategy for deriving the constraints on the GP coefficients of a BF to be β -SD/ASD. For β -self duality, we output complement the function $f(\bar{x})$ and equate it to $f(x)$. Since output complementation merely complements the constant term $\hat{a}_{-\infty}$, keeping the rest of the coefficients invariant, the coefficients of $\bar{f}(\bar{x})$ will be given by $\bar{a}_{-\infty}, \hat{a}_i$, $i = 0, 1, \dots, 2^k-2$, where $\bar{a}_{-\infty}$ is the complement of $\hat{a}_{-\infty}$.

Equating the coefficients of $f(x)$ with those of $\bar{f}(\bar{x})$, we get the conditions for β -self duality

as

$$\hat{a}_1 = a_1,$$

$$\text{ie., } \sum_{t=0}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t a_{1-t} = a_1, i = 0, 1, \dots, 2^k-2$$

$$\text{or } \sum_{t=1}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t a_{1-t} + a_1 = a_1, i = 0, 1, \dots, 2^k-2$$

$$\text{ie., } \sum_{t=1}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t a_{1-t} = 0, i = 1, 2, \dots, 2^k-2 \quad (4.7.1)$$

Secondly,

$$\hat{a}_{-\infty} = a_{-\infty}$$

$$\text{or } \hat{a}_{-\infty} = \bar{a}_{-\infty}$$

$$\text{ie., } \sum_i a_i \beta^{-i} = \bar{a}_{-\infty}, i = -\infty, 0, 1, \dots, 2^k-2.$$

Now since $a_{-\infty} + \bar{a}_{-\infty} = 1$, we may write the above equation as

$$\sum_{i=0}^{\zeta} a_i \beta^{-i} = 1; \zeta = 2^k-2. \quad (4.7.2)$$

Similarly, equating the coefficients of $f(x)$ with those of $f(\bar{x})$, we get the conditions for β -anti self duality as

$$\hat{a}_1 = a_1, i = 0, 1, \dots, 2^k-2.$$

$$\text{or } \sum_{t=1}^i \left[\begin{matrix} i \\ t \end{matrix} \right]_2 \beta^t a_{1-t} = 0, i = 1, 2, \dots, 2^k-2, \quad (4.7.3)$$

$$\text{and } \hat{a}_{-\infty} = a_{-\infty}.$$

$$\text{or } \sum_{i=0}^{\zeta} a_i \beta^{-i} = 0; \zeta = 2^k-2. \quad (4.7.4)$$

Thus for β -self duality, we substitute $\hat{a}_{-\infty}$ and \hat{a}_1 , $i = 0, 1, \dots, 2^k-2$ respectively by $\bar{a}_{-\infty}$

and a_i , $i = 0, 1, \dots, 2^k-2$, and for β -anti self duality we substitute the same by $a_{-\infty}$ and a_i , $i = 0, 1, \dots, 2^k-2$, respectively and then derive the constraints on the coefficients. We also use the conjugacy relations among the coefficients

$$(a_i)^2 = a_{2.i} \quad (4.7.5)$$

(where $2.i$ is taken modulo 2^k-1) wherever required.

4.7.2 Characterization of 2-Variable β -Self Dual / Anti Self Dual Boolean Functions

(1) β -Self Dual Functions

In (4.7.1), substituting $i = 1$, we get

$$\beta a_0 = 0$$

or $a_0 = 0$ since $\beta \neq 0$.

$a_{-\infty}$ can assume 0 or 1.

Using (4.7.2), we have

$$a_0 + \beta^2 a_1 + \beta a_2 = 1.$$

Since $a_0 = 0$, we get

$$\beta^2 a_1 + \beta a_2 = 1.$$

Since a_1 and a_2 are related by conjugacy constraints as

$$a_2 = (a_1)^2,$$

we may write

$$\beta^2 a_1 + (\beta^2 a_1)^2 = 1.$$

In other words, we require a_1 to satisfy the condition

$$\text{frs}(\beta^2 a_1) = 1. \quad (4.7.6)$$

$$\text{Let } \Theta = \beta^2 a_1.$$

Let $x^2 + x + 1$ be a primitive polynomial used for generating $\text{GF}(2^2)$ with γ as a primitive element. Then it may be seen that the elements Θ in $\text{GF}(2^2)$ which satisfy (4.7.6)

are γ and γ^2 . Thus a_1 is given by

$$a_1 = \beta^{-2}\gamma, \beta^{-2}\gamma^2.$$

Summarizing, we have, for β -SD 2 variable BFs, the GP coefficients (one member in each Frobenius cycle) are given by

$$a_{-\infty} = 0 \text{ or } 1. \quad (4.7.7a)$$

$$a_0 = 0. \quad (4.7.7b)$$

and
$$a_1 = \beta^{-2}\gamma, \beta^{-2}\gamma^2. \quad (4.7.7c)$$

This gives a total of 4 β -SD BFs.

We list in Table 4.5, the GP representation of all 2-variable β -SD BFs for all the nonzero values of $\beta \in \text{GF}(2^2)$. We also list the corresponding BFs which are inverse Galois transforms of these coefficients. The coefficients are listed as a power of γ . Only the exponents are listed. The BF is listed in binary form. In all the cases, $a_{-\infty} = 0$ or 1, and $a_0 = 0$.

The coefficient a_1 given by $\beta^{-2}\gamma, \beta^{-2}\gamma^2$, for all nonzero values of β , are as follows:

- | | |
|-----------------------------|----------------------|
| (1) $\beta = 1 = 01$ | γ, γ^2 . |
| (2) $\beta = \gamma = 10$ | $\gamma^2, 1$. |
| (3) $\beta = \gamma^2 = 11$ | $1, \gamma$. |

Table 4.5: 2-Variable β -Self Dual Boolean Functions and their GP Representations

(a) $\beta = 1$

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	1	2	0	1	0	1
2	$-\infty$	$-\infty$	2	1	0	1	1	0
3	0	$-\infty$	1	2	1	0	1	0
4	0	$-\infty$	2	1	1	0	0	1

Table 4.5 (continued)

(b) $\beta = \gamma$

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	2	1	0	1	1	0
2	$-\infty$	$-\infty$	0	0	0	0	1	1
3	0	$-\infty$	2	1	1	0	0	1
4	0	$-\infty$	0	0	1	1	0	0

(c) $\beta = \gamma^2$ (self dual)

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	0	0	0	0	1	1
2	$-\infty$	$-\infty$	1	2	0	1	0	1
3	0	$-\infty$	0	0	1	1	0	0
4	0	$-\infty$	1	2	1	0	1	0

(2) β -Anti Self Dual Functions

For 2-variable β -ASD functions, the only change is in (4.7.4), using which gives the condition

$$\text{frs}(\beta^2 a_1) = 0. \quad (4.7.8)$$

The elements $\Theta = \beta^2 a_1$ in $\text{GF}(2^2)$ which satisfy (4.7.8) are 0 and 1. Thus a_1 is given by

$$a_1 = \beta^{-2}.0, \beta^{-2}.1 = 0, \beta^{-2}.$$

Summarizing, we have, for β -ASD 2 variable BFs, the GP coefficients (one member in each Frobenius cycle) are given by

$$a_{-\infty} = 0 \text{ or } 1. \quad (4.7.9a)$$

$$a_0 = 0. \quad (4.7.9b)$$

and

$$a_1 = 0, \beta^{-2}. \quad (4.7.9c)$$

As in the previous case, we list in Table 4.6, the GP representation of all 2-variable β -ASD BFs for all the nonzero values of $\beta \in \text{GF}(2^2)$. We also list the corresponding BFs

which are inverse Galois transforms of these coefficients. In all the cases, $a_{-\infty} = 0$ or 1 , and $a_0 = 0$.

The coefficient a_1 given by $0, \beta^{-2}$, for all nonzero values of β are as follows:

- (1) $\beta = 1 = 01$ $0, 1$.
- (2) $\beta = \gamma = 10$ $0, \gamma$.
- (3) $\beta = \gamma^2 = 11$ $0, \gamma^2$.

Table 4.6: 2-Variable β -Anti Self Dual Boolean Functions and their GP Representations

(a) $\beta = 1$

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0	0	0
2	$-\infty$	$-\infty$	0	0	0	0	1	1
3	0	$-\infty$	$-\infty$	$-\infty$	1	1	1	1
4	0	$-\infty$	0	0	1	1	0	0

(b) $\beta = \gamma$

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0	0	0
2	$-\infty$	$-\infty$	1	2	0	1	0	1
3	0	$-\infty$	$-\infty$	$-\infty$	1	1	1	1
4	0	$-\infty$	1	2	1	0	1	0

(c) $\beta = \gamma^2$ (anti self dual)

No.	$a_{-\infty}$	a_0	a_1	a_2	Boolean function			
1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0	0	0
2	$-\infty$	$-\infty$	2	1	0	1	1	0
3	0	$-\infty$	$-\infty$	$-\infty$	1	1	1	1
4	0	$-\infty$	2	1	1	0	0	1

4.7.3 Characterization of 3-Variable β -Self Dual / Anti Self Dual Boolean Functions

(1) β -Self Dual Functions

For 3-variable β -SD functions, putting $i = 1$ in (4.7.1), we have

$$\beta a_0 = 0$$

or

$$a_0 = 0 \text{ since } \beta \neq 0.$$

$$a_{-\infty} = 0 \text{ or } 1.$$

Putting $i = 3$ in (4.7.1), we get

$$\beta a_2 + \beta^2 a_1 + \beta^3 a_0 = 0.$$

Substituting $a_0 = 0$, we get

$$\beta^2 a_1 + \beta a_2 = 0$$

or

$$a_2 = \beta a_1$$

Since a_1 and a_2 are related by

$$a_2 = (a_1)^2,$$

we may write

$$(a_1)^2 + \beta a_1 = a_1(a_1 + \beta) = 0.$$

$$\text{Thus } a_1 = 0 \text{ or } \beta.$$

This also fixes a_2 and a_4 , since they are related to a_1 by conjugacy constraints.

To get the constraints on the remaining conjugate set $\{a_3, a_6, a_5\}$, we take (4.7.2).

$$a_0 + \beta^6 a_1 + \beta^5 a_2 + \beta^4 a_3 + \beta^3 a_4 + \beta^2 a_5 + \beta a_6 = 1.$$

Since $a_0 = 0$ and the remaining coefficients satisfy conjugacy relations, we may express the above as a sum of two Frobenius terms, as

$$\text{frs}(\beta^6 a_1) + \text{frs}(\beta^4 a_3) = 1. \quad (4.7.10)$$

Since a_1 assumes only 0 and β , we substitute these values in the above equation to find the corresponding values of a_3 .

When $a_1 = 0$, we get

$$\text{frs}(\beta^4 a_3) = 1$$

When $a_1 = \beta$, we get

$$\text{frs}(\beta^7) + \text{frs}(\beta^4 a_3) = 1.$$

As $\text{frs}(\beta^7) = 1$, for any $\beta \in \text{GF}(2^3)$, we get

$$\text{frs}(\beta^4 a_3) = 0$$

$$\text{Let } \Theta = \beta^4 a_3.$$

Now we are left with choosing those elements Θ of $\text{GF}(2^3)$, whose Frobenius sum is 1, when $a_1 = 0$, and whose Frobenius sum is 0, when $a_1 = \beta$. Half the elements of $\text{GF}(2^3)$ has Frobenius sum equal to 0, and half the elements has Frobenius sum equal to 1. Thus we get a total of 16 β -SD BFs.

To illustrate the case, we choose the primitive polynomial $x^3 + x^2 + 1$ for generating $\text{GF}(2^3)$, with γ as a primitive element. The elements $\Theta \in \text{GF}(2^3)$ whose Frobenius sum is 1 are given by 1, γ , γ^2 and γ^4 , and the elements whose Frobenius sum is 0 are given by 0, γ^3 , γ^6 and γ^5 .

Thus in this case we have, for β -SD 3 variable BFs, the GP coefficients (one member in each Frobenius cycle) are given by

$$a_{-\infty} = 0 \text{ or } 1. \quad (4.7.11a)$$

$$a_0 = 0. \quad (4.7.11b)$$

$$a_1 = 0, \beta. \quad (4.7.11c)$$

$$\begin{aligned} a_3 &= \beta^4, \gamma\beta^4, \gamma^2\beta^4, \gamma^4\beta^4 \\ &= \beta^3, \gamma\beta^3, \gamma^2\beta^3, \gamma^4\beta^3, \text{ when } a_1 = 0, \end{aligned} \quad (4.7.11d)$$

and

$$\begin{aligned} a_3 &= 0, \gamma^3\beta^4, \gamma^6\beta^4, \gamma^5\beta^4. \\ &= 0, \gamma^3\beta^3, \gamma^6\beta^3, \gamma^5\beta^3, \text{ when } a_1 = \beta. \end{aligned} \quad (4.7.11e)$$

Based on the above, we list in Table 4.7, the GP representation of 3-variable β -SD BFs which are completely self dual, ie., β in this case is $111 = \gamma^4$. We also list the corresponding

BFs which are inverse Galois transforms of these coefficients. As before, the coefficients are listed as a power of γ and only the exponents of γ are listed. The BF is listed in binary form. Since, the coefficients are related by conjugacy constraints, only one member of each Frobenius class is listed, namely $a_{-\infty}$, a_0 , a_1 and a_3 , out of a total of 8 coefficients.

Table 4.7: 3-Variable Self Dual Boolean Functions and their GP Representations

No	$a_{-\infty}$	a_0	a_1	a_3	Boolean function
1	$-\infty$	$-\infty$	$-\infty$	0	0 1 1 1 0 1 0 0
2	$-\infty$	$-\infty$	$-\infty$	2	0 1 0 0 1 1 1 0
3	$-\infty$	$-\infty$	$-\infty$	5	0 0 1 0 0 1 1 1
4	$-\infty$	$-\infty$	$-\infty$	6	0 0 0 1 1 1 0 1
5	$-\infty$	$-\infty$	4	$-\infty$	0 1 0 1 1 1 0 0
6	$-\infty$	$-\infty$	4	1	0 0 0 0 1 1 1 1
7	$-\infty$	$-\infty$	4	3	0 1 1 0 0 1 1 0
8	$-\infty$	$-\infty$	4	4	0 0 1 1 0 1 0 1
9	0	$-\infty$	$-\infty$	0	1 0 0 0 1 0 1 1
10	0	$-\infty$	$-\infty$	2	1 0 1 1 0 0 0 1
11	0	$-\infty$	$-\infty$	5	1 1 0 1 1 0 0 0
12	0	$-\infty$	$-\infty$	6	1 1 1 0 0 0 1 0
13	0	$-\infty$	4	$-\infty$	1 0 1 0 0 0 1 1
14	0	$-\infty$	4	1	1 1 1 1 0 0 0 0
15	0	$-\infty$	4	3	1 0 0 1 1 0 0 1
16	0	$-\infty$	4	4	1 1 0 0 1 0 1 0

(2) β -Anti Self Dual Functions

For 3-variable β -ASD functions, we get the same conditions on $a_{-\infty}$, a_0 and

$\{a_1, a_2, a_4\}$, as in the case of β -SD functions.

ie., $a_0 = 0$

$a_{-\infty}$ assumes 0 or 1,

and a_1 assumes either 0 or β . Only the constraint on the remaining conjugate set

$\{a_3, a_6, a_5\}$ is changed.

$$\text{frs}(\beta^6 a_1) + \text{frs}(\beta^4 a_3) = 0. \quad (4.7.12)$$

When $a_1 = 0$, we get

$$\text{frs}(\beta^4 a_3) = 0$$

When $a_1 = \beta$, we get

$$\text{frs}(\beta^7) + \text{frs}(\beta^4 a_3) = 0$$

$$\text{frs}(\beta^4 a_3) = 1, \text{ as } \text{frs}(\beta^7) = 1.$$

$$\text{Let } \Theta = \beta^4 a_3.$$

Thus we choose those elements Θ of $\text{GF}(2^3)$, whose Frobenius sum is 0, when $a_1 = 0$, and whose Frobenius sum is 1, when $a_1 = \beta$. Thus we get a total of 16 β -ASD BFs.

As before, we choose the primitive polynomial $x^3 + x^2 + 1$ for generating $\text{GF}(2^3)$, with γ as a primitive element.

Thus we have, for 3 variable β -ASD BFs, the GP coefficients (one member in each Frobenius cycle) are given by

$$a_{-\infty} = 0 \text{ or } 1. \quad (4.7.13a)$$

$$a_0 = 0. \quad (4.7.13b)$$

$$a_1 = 0, \beta. \quad (4.7.13c)$$

$$\begin{aligned} a_3 &= 0, \gamma^3 \beta^4, \gamma^6 \beta^4, \gamma^5 \beta^4. \\ &= 0, \gamma^3 \beta^3, \gamma^6 \beta^3, \gamma^5 \beta^3, \text{ when } a_1 = 0. \end{aligned} \quad (4.7.13d)$$

and

$$\begin{aligned} a_3 &= \beta^4, \gamma \beta^4, \gamma^2 \beta^4, \gamma^4 \beta^4. \\ &= \beta^3, \gamma \beta^3, \gamma^2 \beta^3, \gamma^4 \beta^3, \text{ when } a_1 = \beta \end{aligned} \quad (4.7.13e)$$

Based on the above, we list in Table 4.8, the GP representation of all 3-variable completely ASD BFs. We also list the corresponding BFs by finding the inverse Galois transforms of these coefficients. As before, only one member of each Frobenius class is listed, namely $a_{-\infty}$, a_0 , a_1 and a_3 , out of a total of 8 coefficients.

Table 4.8: 3-Variable Anti Self Dual Boolean Functions and their GP Representations

No	$a_{-\infty}$	a_0	a_1	a_3	Boolean function
1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	00000000
2	$-\infty$	$-\infty$	$-\infty$	1	01010011
3	$-\infty$	$-\infty$	$-\infty$	3	00111010
4	$-\infty$	$-\infty$	$-\infty$	4	01101001
5	$-\infty$	$-\infty$	4	0	00101000
6	$-\infty$	$-\infty$	4	2	00010010
7	$-\infty$	$-\infty$	4	5	01111011
8	$-\infty$	$-\infty$	4	6	01000001
9	0	$-\infty$	$-\infty$	$-\infty$	11111111
10	0	$-\infty$	$-\infty$	1	10101100
11	0	$-\infty$	$-\infty$	3	11000101
12	0	$-\infty$	$-\infty$	4	10010110
13	0	$-\infty$	4	0	11010111
14	0	$-\infty$	4	2	11101101
15	0	$-\infty$	4	5	10000100
16	0	$-\infty$	4	6	10111110

4.7.4 Characterization of 4-Variable β -Self Dual / Anti Self Dual Boolean Functions

(1) β -Self Dual Functions

We get constraints for one member of each Frobenius class as before. Thus for 4-variable β -SD functions, putting $i = 1$ in (4.7.1), we have

$$\beta a_0 = 0$$

or

$$a_0 = 0, \text{ since } \beta \neq 0.$$

$$a_{-\infty} \text{ can assume 0 or 1.}$$

Putting $i = 3$ in (4.7.1), we get

$$\beta a_2 + \beta^2 a_1 + \beta^3 a_0 = 0.$$

Substituting $a_0 = 0$, and $a_2 = (a_1)^2$ we get

$$(a_1)^2 + \beta a_1 = a_1(a_1 + \beta) = 0.$$

Thus a_1 assumes either 0 or β .

Putting $i = 7$ in (4.7.1) and substituting $a_0 = 0$, we get

$$\beta a_6 + \beta^2 a_5 + \beta^3 a_4 + \beta^4 a_3 + \beta^5 a_2 + \beta^6 a_1 = 0. \quad (4.7.14)$$

Multiplying (4.7.14) by β^8 , we get

$$\beta^9 a_6 + \beta^{10} a_5 + \beta^{11} a_4 + \beta^{12} a_3 + \beta^{13} a_2 + \beta^{14} a_1 = 0. \quad (4.7.15)$$

(4.7.15) may be written as

$$\text{frs}(\beta^{14} a_1) + (\beta^{12} a_3) + (\beta^{12} a_3)^2 + \beta^7 a_8 + \beta^{10} a_5 = 0 \quad (4.7.16)$$

Let $B = \beta^{12} a_3$.

When $a_1 = 0$, $a_8 = 0$, $\beta^7 a_8 = 0$, $\text{frs}(\beta^{14} a_1) = 0$, and (4.7.16) becomes

$$B + B^2 + \beta^{10} a_5 = 0 \quad (4.7.17)$$

When $a_1 = \beta$, $a_8 = \beta^8$, $\beta^7 a_8 = 1$, $\text{frs}(\beta^{14} a_1) = 0$, and (4.7.16) becomes

$$B + B^2 + \beta^{10} a_5 = 1 \quad (4.7.18)$$

Now let us put $i = 14$ in (4.7.1) and substitute $a_0 = 0$, we get

$$\beta^2 a_{12} + \beta^4 a_{10} + \beta^6 a_8 + \beta^8 a_6 + \beta^{10} a_4 + \beta^{12} a_2 = 0. \quad (4.7.19)$$

Multiplying (4.7.19) by β , we get

$$\beta^3 a_{12} + \beta^5 a_{10} + \beta^7 a_8 + \beta^9 a_6 + \beta^{11} a_4 + \beta^{13} a_2 = 0. \quad (4.7.20)$$

Adding (4.7.15) and (4.7.20) gives

$$\text{frs}(\beta^{10} a_5) + B + B^4 = \beta^{14} a_1 + \beta^7 a_8. \quad (4.7.21)$$

The RHS of (4.7.21) is zero, for $a_1 = 0, \beta$.

Thus we get

$$\text{frs}(\beta^{10} a_5) = B + B^4. \quad (4.7.22)$$

Putting $i = 13$ in (4.7.1) and substituting $a_0 = 0$, we get

$$\beta a_{12} + \beta^4 a_9 + \beta^5 a_8 + \beta^8 a_5 + \beta^9 a_4 + \beta^{12} a_1 = 0. \quad (4.7.23)$$

Multiplying (4.7.23) by β^2 , we get

$$\beta^3 a_{12} + \beta^6 a_9 + \beta^7 a_8 + \beta^{10} a_5 + \beta^{11} a_4 + \beta^{14} a_1 = 0. \quad (4.7.24)$$

Adding (4.7.15) and (4.7.24) gives

$$\text{frs}(\beta^{12} a_3) = \beta^{13} a_2 + \beta^7 a_8. \quad (4.7.25)$$

The RHS of (4.7.25) is zero, for $a_1 = 0, \beta$.

Thus we get

$$\text{frs}(\beta^{12} a_3) = 0. \quad (4.7.26)$$

Now we use (4.7.2) and substitute $a_0 = 0$, to get

$$\text{frs}(\beta^{14} a_1) + \text{frs}(\beta^{12} a_3) + \text{frs}(\beta^{10} a_5) + \text{frs}(\beta^8 a_7) = 1.$$

Since the first two terms are zero, we get

$$\text{frs}(\beta^{10} a_5) + \text{frs}(\beta^8 a_7) = 1. \quad (4.7.27)$$

Substituting (4.7.22) in (4.7.27), we get

$$B + B^4 + \text{frs}(\beta^8 a_7) = 1. \quad (4.7.28)$$

Thus Equations (4.7.17), (4.7.18), (4.7.26) and (4.7.28) give the conditions for β -self duality in the case of 4-variable BFs.

We list in Table 4.9, all 4-variable (completely) SD BFs, both in terms of its GP coefficients and the corresponding functions in binary form. We choose the primitive polynomial for $\text{GF}(2^4)$ as $x^4 + x + 1$ with γ as a primitive element. Only one coefficient from each Frobenius class is listed. There are 256 4-variable SD BFs. To save space, we list only half of them in Table 4.9, corresponding to $a_{-\infty} = \gamma^{-\infty} = 0$. The remaining half consists of functions which are complements of the first half and can be found by just replacing $a_{-\infty} = \gamma^0 = 1$, keeping the remaining coefficients unchanged. Since $a_{-\infty}$ and a_0 are 0 throughout in the listing, we do not list them. Thus only a_1, a_3, a_5 and a_7 are listed. Here $\beta = 1111 = \gamma^{12}$.

Table 4.9: 4-Variable Self Dual Boolean Functions and their GP Representations

No.	a_1	a_3	a_5	a_7	Boolean function
1	∞	∞	∞	0	0000100110101111
2	∞	∞	∞	1	0010011010111100
3	∞	∞	∞	3	0110101111000100
4	∞	∞	∞	5	0011110001001101
5	∞	∞	∞	6	0111000100110101
6	∞	∞	∞	7	0100010011010111
7	∞	∞	∞	8	0001001101011110
8	∞	∞	∞	12	0101111000100110
9	∞	1	0	0	0001011000111110
10	∞	1	0	1	0011100100101101
11	∞	1	0	3	0111010001010101
12	∞	1	0	5	0010001111011100
13	∞	1	0	6	0110111010100100
14	∞	1	0	7	0101101101000110
15	∞	1	0	8	0000110011001111
16	∞	1	0	12	0100000110110111
17	∞	6	∞	0	0110011011010100
18	∞	6	∞	1	0100100111000111
19	∞	6	∞	3	0000010010111111
20	∞	6	∞	5	0101001100110110
21	∞	6	∞	6	0001111001001110
22	∞	6	∞	7	0010101110101100
23	∞	6	∞	8	0111110000100101
24	∞	6	∞	12	0011000101011101
25	∞	7	5	∞	0001000100111111
26	∞	7	5	2	0000101111001110
27	∞	7	5	4	0011111000101100
28	∞	7	5	9	0101110001000111
29	∞	7	5	10	0010010011011101
30	∞	7	5	11	0100011010110110
31	∞	7	5	13	0110100110100101
32	∞	7	5	14	0111001101010100
33	∞	8	10	∞	0110000111010101
34	∞	8	10	2	0111101100100100
35	∞	8	10	4	0100111011000110
36	∞	8	10	9	0010110010101101
37	∞	8	10	10	0101010000110111
38	∞	8	10	11	0011011001011100
39	∞	8	10	13	0001100101001111
40	∞	8	10	14	0000001110111110
41	∞	10	5	∞	0111111001000100
42	∞	10	5	2	0110010010110101
43	∞	10	5	4	0101000101010111
44	∞	10	5	9	0011001100111100
45	∞	10	5	10	0100101110100110
46	∞	10	5	11	0010100111001101

Table 4.9 (continued)

No.	a_1	a_3	a_5	a_7	Boolean function
47	∞	10	5	13	0000011011011110
48	∞	10	5	14	0001110000101111
49	∞	11	0	0	0111100101000101
50	∞	11	0	1	0101011001010110
51	∞	11	0	3	0001101100101110
52	∞	11	0	5	0100110010100111
53	∞	11	0	6	0000000111011111
54	∞	11	0	7	0011010000111101
55	∞	11	0	8	0110001110110100
56	∞	11	0	12	0010111011001100
57	∞	14	10	∞	0000111010101110
58	∞	14	10	2	0001010001011111
59	∞	14	10	4	0010000110111101
60	∞	14	10	9	0100001111010110
61	∞	14	10	10	0011101101001100
62	∞	14	10	11	0101100100100111
63	∞	14	10	13	0111011000110100
64	∞	14	10	14	0110110011000101
65	12	∞	0	0	0101010001010111
66	12	∞	0	1	0111101101000100
67	12	∞	0	3	0011011000111100
68	12	∞	0	5	0110000110110101
69	12	∞	0	6	0010110011001101
70	12	∞	0	7	0001100100101111
71	12	∞	0	8	0100111010100110
72	12	∞	0	12	0000001111011110
73	12	1	∞	0	0100101111000110
74	12	1	∞	1	0110010011010101
75	12	1	∞	3	0010100110101101
76	12	1	∞	5	0111111000100100
77	12	1	∞	6	0011001101011100
78	12	1	∞	7	0000011010111110
79	12	1	∞	8	0101000100110111
80	12	1	∞	12	0001110001001111
81	12	6	0	0	0011101100101100
82	12	6	0	1	0001010000111111
83	12	6	0	3	0101100101000111
84	12	6	0	5	0000111011001110
85	12	6	0	6	0100001110110110
86	12	6	0	7	0111011001010100
87	12	6	0	8	0010000111011101
88	12	6	0	12	0110110010100101
89	12	7	10	∞	0100110011000111
90	12	7	10	2	0101011000110110
91	12	7	10	4	0110001111010100
92	12	7	10	9	0000000110111111
93	12	7	10	10	0111100100100101
94	12	7	10	11	0001101101001110
95	12	7	10	13	0011010001011101

Table 4.9 (continued)

No.	a_1	a_3	a_5	a_7	Boolean function
96	12	7	10	14	0010111010101100
97	12	8	5	$-\infty$	0011110000101101
98	12	8	5	2	0010011011011100
99	12	8	5	4	0001001100111110
100	12	8	5	9	0111000101010101
101	12	8	5	10	0000100111001111
102	12	8	5	11	0110101110100100
103	12	8	5	13	0100010010110111
104	12	8	5	14	0101111001000110
105	12	10	10	$-\infty$	0010001110111100
106	12	10	10	2	0011100101001101
107	12	10	10	4	0000110010101111
108	12	10	10	9	0110111011000100
109	12	10	10	10	0001011001011110
110	12	10	10	11	0111010000110101
111	12	10	10	13	0101101100100110
112	12	10	10	14	0100000111010111
113	12	11	$-\infty$	0	0010010010111101
114	12	11	$-\infty$	1	0000101110101110
115	12	11	$-\infty$	3	0100011011010110
116	12	11	$-\infty$	5	0001000101011111
117	12	11	$-\infty$	6	0101110000100111
118	12	11	$-\infty$	7	0110100111000101
119	12	11	$-\infty$	8	0011111001001100
120	12	11	$-\infty$	12	0111001100110100
121	12	14	5	$-\infty$	0101001101010110
122	12	14	5	2	0100100110100111
123	12	14	5	4	0111110001000101
124	12	14	5	9	0001111000101110
125	12	14	5	10	0110011010110100
126	12	14	5	11	0000010011011111
127	12	14	5	13	0010101111001100
128	12	14	5	14	0011000100111101

(2) β -Anti Self Dual Functions

For 4-variable β -ASD functions, we get the same conditions on $a_{-\infty}$, a_0 , and a_1 as in the case of β -SD functions. Further, the conditions (4.7.17), (4.7.18) and (4.7.26) also remain unchanged. But (4.7.28) gets modified to

$$B + B^4 + \text{frs}(\beta^8 a_7) = 0 \quad (4.7.29)$$

We list in Table 4.10, all 4-variable (completely) ASD BFs, both in terms of its GP

coefficients and the corresponding functions in binary form. As before, we choose the primitive polynomial for $GF(2^4)$ as $x^4 + x + 1$ with γ as a primitive element. Out of the 256 4-variable ASD BFs, only half of them is listed, the remaining being complements of this half. $a_{-\infty}$ and a_0 are not listed, being 0 throughout. $\beta = 1111 = \gamma^{12}$.

Table 4.10: 4-Variable Anti Self Dual Boolean Functions and their GP Representations

	a_1	a_3	a_5	a_7	Boolean function
1	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0000000000000000
2	$-\infty$	$-\infty$	$-\infty$	2	0001101011110001
3	$-\infty$	$-\infty$	$-\infty$	4	0010111100010011
4	$-\infty$	$-\infty$	$-\infty$	9	0100110101111000
5	$-\infty$	$-\infty$	$-\infty$	10	0011010111100010
6	$-\infty$	$-\infty$	$-\infty$	11	0101011110001001
7	$-\infty$	$-\infty$	$-\infty$	13	0111100010011010
8	$-\infty$	$-\infty$	$-\infty$	14	0110001001101011
9	$-\infty$	1	0	$-\infty$	0001111110010001
10	$-\infty$	1	0	2	0000010101100000
11	$-\infty$	1	0	4	0011000010000010
12	$-\infty$	1	0	9	0101001011101001
13	$-\infty$	1	0	10	0010101001110011
14	$-\infty$	1	0	11	0100100000011000
15	$-\infty$	1	0	13	0110011100001011
16	$-\infty$	1	0	14	0111110111111010
17	$-\infty$	6	$-\infty$	$-\infty$	0110111101111011
18	$-\infty$	6	$-\infty$	2	0111010110001010
19	$-\infty$	6	$-\infty$	4	0100000001101000
20	$-\infty$	6	$-\infty$	9	0010001000000011
21	$-\infty$	6	$-\infty$	10	0101101010011001
22	$-\infty$	6	$-\infty$	11	0011100011110010
23	$-\infty$	6	$-\infty$	13	0001011111100001
24	$-\infty$	6	$-\infty$	14	0000110100010000
25	$-\infty$	7	5	0	0001100010010000
26	$-\infty$	7	5	1	0011011110000011
27	$-\infty$	7	5	3	0111101011111011
28	$-\infty$	7	5	5	00101101011110010
29	$-\infty$	7	5	6	0110000000001010
30	$-\infty$	7	5	7	0101010111101000
31	$-\infty$	7	5	8	0000001001100001
32	$-\infty$	7	5	12	0100111100011001
33	$-\infty$	8	10	0	0110100001111010
31	$-\infty$	7	5	8	0000001001100001
32	$-\infty$	7	5	12	0100111100011001
33	$-\infty$	8	10	0	0110100001111010

Table 4.10 (continued)

	a_1	a_3	a_5	a_7	Boolean function
34	∞	8	10	1	0100011101101001
35	∞	8	10	3	0000101000010001
36	∞	8	10	5	0101110110011000
37	∞	8	10	6	0001000011100000
38	∞	8	10	7	0010010100000010
39	∞	8	10	8	0111001010001011
40	∞	8	10	12	0011111111110011
41	∞	10	5	0	0111011111101011
42	∞	10	5	1	0101100011111000
43	∞	10	5	3	0001010110000000
44	∞	10	5	5	0100001000001001
45	∞	10	5	6	0000111101110001
46	∞	10	5	7	0011101010010011
47	∞	10	5	8	0110110100011010
48	∞	10	5	12	0010000001100010
49	∞	11	0	∞	0111000011101010
50	∞	11	0	2	0110101000011011
51	∞	11	0	4	0101111111111001
52	∞	11	0	9	0011110110010010
53	∞	11	0	10	0100010100001000
54	∞	11	0	11	0010011101100011
55	∞	11	0	13	0000100001110000
56	∞	11	0	14	0001001010000001
57	∞	14	10	0	0000011100000001
58	∞	14	10	1	0010100000010010
59	∞	14	10	3	0110010101101010
60	∞	14	10	5	0011001011100011
61	∞	14	10	6	0111111110011011
62	∞	14	10	7	0100101001111001
63	∞	14	10	8	0001110111110000
64	∞	14	10	12	0101000010001000
65	12	∞	0	∞	0101110111111000
66	12	∞	0	2	0100011100001001
67	12	∞	0	4	0111001011101011
68	12	∞	0	9	0001000010000000
69	12	∞	0	10	0110100000011010
70	12	∞	0	11	0000101001110001
71	12	∞	0	13	0010010101100010
72	12	∞	0	14	0011111110010011
73	12	1	∞	∞	0100001001101001
74	12	1	∞	2	0101100010011000
75	12	1	∞	4	0110110101111010
76	12	1	∞	9	0000111100010001
77	12	1	∞	10	0111011110001011
78	12	1	∞	11	0001010111100000
79	12	1	∞	13	0011101011110011
80	12	1	∞	14	0010000000000010
81	12	6	0	∞	0011001010000011

Table 4.10 (continued)

	a_1	a_3	a_5	a_7	Boolean function
82	12	6	0	2	0010100001110010
83	12	6	0	4	0001110110010000
84	12	6	0	9	011111111111011
85	12	6	0	10	0000011101100001
86	12	6	0	11	0110010100001010
87	12	6	0	13	0100101000011001
88	12	6	0	14	0101000011101000
89	12	7	10	0	0100010101101000
90	12	7	10	1	0110101001111011
91	12	7	10	3	0010011100000011
92	12	7	10	5	0111000010001010
93	12	7	10	6	0011110111110010
94	12	7	10	7	0000100000010000
95	12	7	10	8	010111110011001
96	12	7	10	12	0001001011100001
97	12	8	5	0	0011010110000010
98	12	8	5	1	0001101010010001
99	12	8	5	3	0101011111101001
100	12	8	5	5	000000001100000
101	12	8	5	6	0100110100011000
102	12	8	5	7	0111100011111010
103	12	8	5	8	0010111101110011
104	12	8	5	12	0110001000001011
105	12	10	10	0	0010101000010011
106	12	10	10	1	0000010100000000
107	12	10	10	3	0100100001111000
108	12	10	10	5	0001111111110001
109	12	10	10	6	0101001010001001
110	12	10	10	7	0110011101101011
111	12	10	10	8	0011000011100010
112	12	10	10	12	0111110110011010
113	12	11	$\neg a$	$\neg a$	0010110100010010
114	12	11	$\neg a$	2	0011011111100011
115	12	11	$\neg a$	4	0000001000000001
116	12	11	$\neg a$	9	0110000001101010
117	12	11	$\neg a$	10	0001100011110000
118	12	11	$\neg a$	11	0111101010011011
119	12	11	$\neg a$	13	0101010110001000
120	12	11	$\neg a$	14	0100111101111001
121	12	14	5	0	0101101011111001
122	12	14	5	1	0111010111101010
123	12	14	5	3	0011100010010010
124	12	14	5	5	0110111100011011
125	12	14	5	6	0010001001100011
126	12	14	5	7	0001011110000001
127	12	14	5	8	0100000000001000
128	12	14	5	12	0000110101110000

CHAPTER 5

GSF THEORY FOR ERROR CONTROL CODES

In this chapter, we consider applications of GSFs in coding theory and techniques. We have shown in Chapter 3, that there is a one-to-one correspondence between linear (n,k) transformations and linearized GSFs (LGSFs) of a given pair of n and k . Since, a linear (n,k) block code is a linear (n,k) transformation which represents a one-to-one linear mapping, naturally all linear (n,k) block codes are *linearized* GSFs which can be represented by appropriate linearized Galois polynomials (LGPs). Such representations are analogous to the generator matrix (basis) representations of the same. Thus each linear block code has different LGP representations according to the number of ways in which a basis can be chosen for the same. Advantages of different polynomial computation techniques may be exploited in the case of LGP representations, for encoding and decoding of linear block codes, since encoding and decoding operations now reduce to mere polynomial computations. In case of linear (n,k) block codes whose LGP representations have nontrivial conjugacy relations among their coefficients, the encoder structure reduces to that of a Frobenius sum computer which can efficiently compute polynomial values if normal basis (NB) is employed.

Since all LGSFs do not represent linear (n,k) codes, we derive conditions for a LGSF to represent a one-to-one mapping or a linear (n,k) block code and show that for a LGSF to represent a linear block code, the coefficients of its LGP representation have to satisfy certain nonzero determinant property.

In this chapter, we also study classes of LGSFs in terms of the nature of the linear transformations generated by them and show that if one function in such a class represents

near mapping which is one-to-one (many-to-one), then others in that class also represent linear mappings which are one-to-one (many-to-one). Further, a study of single term LGPs is attempted to show that they always represent one-to-one mappings when k divides n . A study of the distinctness of the codes generated by single term LGPs which are members of a finite field (the algebraic structure of such classes was discussed in Chapter 2) is conducted and the number of distinct codes in each field is computed.

A study of the roots of LGPs representing linear (n,k) block codes is carried out.

Canonic LGP representations of cyclic codes are derived both in the standard basis and in normal basis (NB).

Role of GSFs in the decoding of linear (n,k) block codes is considered and a variety of techniques for the standard array decoding of linear block codes are proposed using one-dimensional (1-D) as well as two-dimensional (2-D) GSFs. In the 1-D case, it is shown that all linear (n,k) block codes can have decoders which can be constructed as a sum of various Frobenius sum computers, if the received n -tuple vector is decoded directly to the corresponding k -tuple message vector. Thus besides, employing NB representations for implementation purposes, the fact that each Frobenius term in the LGP which performs the decoding is independently realizable, enables a parallel implementation scheme for the decoder for fast decoding of the respective block codes.

In the following sections, as in Chapter 3, when the coefficients exhibit nontrivial conjugacy relations, the corresponding linearized Galois polynomial (LGP) will be called a linearized Frobenius polynomial (LFP) whereas the same will be called simply as a linearized polynomial (LP) if the conjugacy relations are trivial. When a general reference is made which includes both, the corresponding polynomial will be called simply a LGP. Functions representing the corresponding mappings will be respectively called linearized Frobenius functions (LFFs), linearized functions (LFs) and linearized GSFs (LGSFs). Further, the term GSF is used for one-dimensional (1-D) GSFs unless otherwise stated.

Our discussions are limited to codes over $GF(2)$. Thus the term 'Linear Code' implies a binary linear (n,k) block code wherever it is used, unless otherwise stated.

Another fact which is to be recalled is the interpretation of the LGP coefficients as the Galois spectrum. In other words, the coefficients of the LGP are in fact the Galois transform (GT) coefficients of a signal vector of length 2^k over $GF(2^n)$.

1 Representation of Linear Codes by Linearized GPs

Since a linear (n,k) code is a special case of a linear (n,k) transformation which presents a one-to-one mapping, it may be represented by a LGP of the form (3.1.3). Thus we state the following theorem without proof:

Theorem 5.1.1: A linear (n,k) code over $GF(2)$ with block length n and dimension k , can be represented by a LGP of the form (3.1.3) given by

$$f(x) = \sum_{i=0}^{k-1} a_{-2^i} x^{2^i},$$

where $f(x) \in GF(2^n)$, $x \in GF(2^k)$, $a_{-2^i} \in GF(2^L)$, $i = 0, 1, \dots, k-1$,

and L is the L.C.M. of n and k .

5.1.1 Number of Linearized GPs Representing Linear Codes

We have shown in Chapter 3 that the coefficients of the LGP representing a linear (n,k) transformation are related to the set of vectors chosen for generating the same. For a linear (n,k) code, naturally this set of vectors becomes a basis for the code consisting of k linearly independent vectors (called the *generator matrix* of the code). Thus each code will have as many number of different LGP representations as the number of ways in which a basis can be chosen for the code. The number of ways in which a k -dimensional basis can be chosen for a linear (n,k) code was given by N_k in (3.5.2). Thus one linear (n,k) code has N_k different LGP representations and there are $N_{\text{dist}} = N_n / N_k$ number of distinct linear

codes for a given pair of n and k , according to the arguments given in Section 3.5.3.

Therefore the number of LGPs representing linear (n,k) codes, for a given pair of n and k , is N_n , including the different representations of the same code. But the total number of LGPs for the same n and k , (equal to the total number of possible linear (n,k) transformations), is equal to $2^{n \cdot k}$, which is greater than N_n .

5.2 Condition for Linearized GPs to Represent Linear Codes

We saw in Section 5.1.1 that, out of the $2^{n \cdot k}$ possible LGPs, only N_n LGPs represent one-to-one mappings or linear (n,k) codes. Therefore, in this section, we will derive the constraint on the coefficients of a LGP to represent a linear (n,k) code. This is stated in the following theorem:

Theorem 5.2.1: A LGSF, mapping from $GF(2^k)$ to $GF(2^n)$, described by a LGP

$f(x) = \sum_{i=0}^{k-1} \gamma_i x^{2^i}$, represents a linear (n,k) code iff the coefficients γ_i , $i = 0, 1, \dots, k-1$,

which belong to $GF(2^L)$, L being the L.C.M. of n and k , satisfy the condition

$$\det \begin{vmatrix} \gamma_0 & \gamma_{k-1}^2 & \gamma_{k-2}^{2^2} & \cdot & \cdot & \gamma_1^{2^{k-1}} \\ \gamma_1 & \gamma_0^2 & \gamma_{k-1}^{2^2} & \cdot & \cdot & \gamma_2^{2^{k-1}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \gamma_{k-1} & \gamma_{k-2}^2 & \gamma_{k-3}^{2^2} & \cdot & \cdot & \gamma_0^{2^{k-1}} \end{vmatrix} \neq 0, \quad (5.2.1)$$

where $\det | \cdot |$ is the determinant of the corresponding matrix.

Proof: Raising $f(x)$ to the power 2^j on both sides, for $j = 0, 1, \dots, k-1$,

$$(f(x))^{2^j} = \left(\sum_{i=0}^{k-1} \gamma_i x^{2^i} \right)^{2^j} = \sum_{i=0}^{k-1} \gamma_i^{2^j} x^{2^{i+j}}, \quad j = 0, 1, \dots, k-1. \quad (5.2.2)$$

Let $f(\alpha_s) = \beta_s \in \text{GF}(2^n)$ for $s = 0, 1, \dots, k-1$.

Substituting in (5.2.2), we get,

$$(\beta_s)^{2^j} = \sum_{i=0}^{k-1} \gamma_i^{2^j} \alpha_s^{2^{i+j}}, \quad 0 \leq s, j \leq k-1. \quad (5.2.3a)$$

Since $\alpha_s \in \text{GF}(2^k)$, $\alpha_s^{2^k} = \alpha_s$.

Thus (5.2.3a) can be modified as

$$(\beta_s)^{2^j} = \sum_{i=0}^{k-1} \alpha_s^{2^i} \gamma_{(i-j) \bmod k}^{2^j}, \quad 0 \leq s, j \leq k-1. \quad (5.2.3b)$$

Thus for $s = 0, j = 0$

$$(\beta_0) = \sum_{i=0}^{k-1} \alpha_0^{2^i} \gamma_{i \bmod k}$$

for $s = 0, j = 1$

$$(\beta_0)^2 = \sum_{i=0}^{k-1} \alpha_0^{2^i} \gamma_{(i-1) \bmod k}^2$$

for $s = 0, j = k-1$

$$(\beta_0)^{2^{k-1}} = \sum_{i=0}^{k-1} \alpha_0^{2^i} \gamma_{(i-k+1) \bmod k}^{2^{k-1}}$$

Similarly, expressions can be written for $s = 1, 2, \dots, k-1, 0 \leq j \leq k-1$.

This can be put in matrix form as

$$\underline{\Delta}_2 = \underline{\Delta}_1 \underline{A}. \quad (5.2.4)$$

Where $\underline{\Delta}_2 =$
$$\begin{bmatrix} \beta_0 & \beta_0^2 & \beta_0^{2^2} & . & . & \beta_0^{2^{k-1}} \\ \beta_1 & \beta_1^2 & \beta_1^{2^2} & . & . & \beta_1^{2^{k-1}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \beta_{k-1} & \beta_{k-1}^2 & \beta_{k-1}^{2^2} & . & . & \beta_{k-1}^{2^{k-1}} \end{bmatrix} \quad (5.2.5a)$$

$$\underline{\Delta}_1 = \begin{bmatrix} \alpha_0 & \alpha_0^2 & \alpha_0^{2^2} & . & . & \alpha_0^{2^{k-1}} \\ \alpha_1 & \alpha_1^2 & \alpha_1^{2^2} & . & . & \alpha_1^{2^{k-1}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \alpha_{k-1} & \alpha_{k-1}^2 & \alpha_{k-1}^{2^2} & . & . & \alpha_{k-1}^{2^{k-1}} \end{bmatrix} \quad (5.2.5b)$$

and

$$\underline{A} = \begin{bmatrix} \gamma_0 & \gamma_{k-1}^2 & \gamma_{k-2}^{2^2} & . & . & \gamma_1^{2^{k-1}} \\ \gamma_1 & \gamma_0^2 & \gamma_{k-1}^{2^2} & . & . & \gamma_2^{2^{k-1}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \gamma_{k-1} & \gamma_{k-2}^2 & \gamma_{k-3}^{2^2} & . & . & \gamma_0^{2^{k-1}} \end{bmatrix} \quad (5.2.5c)$$

Since $\underline{\Delta}_2 = \underline{\Delta}_1 \underline{A}$, we can write the corresponding determinant as

$$\det \underline{\Delta}_2 = \det \underline{\Delta}_1 \cdot \det \underline{A}. \quad (5.2.6)$$

Now $\det \underline{\Delta}_1$ and $\det \underline{\Delta}_2 \neq 0$ if and only if $\alpha_s \in \text{GF}(2^k)$, $\beta_s \in \text{GF}(2^n)$, $s = 0, 1, \dots, k-1$, respectively are linearly independent [27].

Since, in our case, α_s , $s = 0, 1, \dots, k-1$, are the SB vectors of $GF(2^k)$, they are always linearly independent and hence $\det \underline{\Delta}_1 \neq 0$. Hence $\det \underline{\Delta}_2 \neq 0$, iff $\det \underline{A} \neq 0$, in which case, $\beta_s \in GF(2^n)$, $s = 0, 1, \dots, k-1$, are linearly independent and hence can form the basis vectors of a linear (n, k) code. Q.E.D.

Corollary 5.2.1: $\det \underline{A}$ belongs to $GF(2^n)$.

Proof: The determinant of matrices of the form of $\underline{\Delta}_1$ and $\underline{\Delta}_2$ is given by the relation

$$\det = b_1 \prod_{j=1}^{k-1} \prod_{c_1 c_2 \dots c_j} (b_{j+1} - \sum_{t=1}^j c_t b_t), \quad (5.2.7)$$

where c_i , $i = 1, 2, \dots, j$, $\in GF(2)$, and b_i , $i = 1, 2, \dots, k$, are the elements of the first column [27]. Substituting b_i 's as the k SB vectors of $GF(2^k)$, namely, $1, \alpha, \alpha^2, \dots, \alpha^{k-1}$, we get the value of $\det \underline{\Delta}_1$ as a product of all the nonzero elements of $GF(2^k)$, which is equal to 1. Substituting in (5.2.6), we get,

$$1 \cdot \det \underline{A} = \det \underline{\Delta}_2.$$

$$\text{or } \det \underline{A} = \det \underline{\Delta}_2. \quad (5.2.8)$$

$$\text{Since } \det \underline{\Delta}_2 \in GF(2^n), \det \underline{A} \in GF(2^n). \quad \text{Q.E.D.}$$

Corollary 5.2.2: $\det \underline{A}$ is the same for different LGPs representing the same code.

Proof: Since, $\det \underline{A} = \det \underline{\Delta}_2$, we can use the same expression for determinant in (5.1.7), to compute $\det \underline{A}$. In this case, the b_i 's are the basis vectors of the corresponding linear code. On substituting b_i 's as the k basis vectors of the linear code, namely, $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$, we get the value of $\det \underline{\Delta}_2$ as a product of β_i , $i = 0, 1, \dots, k-1$, and their linear combinations. In other words, we get the determinant as a product of all the nonzero code vectors of the code. Thus the determinant will be the same irrespective of the basis chosen for the code. Hence $\det \underline{A}$ will be same for the different LGP representations of the same

code.

Q.E.D.

Note: From Corollary 5.2.2, it follows that if $\det \underline{A}$ is different for two LGPs, then they cannot represent the same code. However, if $\det \underline{A}$ is same for two different LGPs, this does not imply that they represent the same code. In other words, two different codes can have the same value of $\det \underline{A}$, as the number of distinct codes can be greater than the number of values which $\det \underline{A}$ can assume, ie., 2^n .

5.3 Representation of Classes of Linear Codes of the Same Weight Distribution by Linearized GPs

In this section, we examine the nature of the linear transformations generated by

LGP's of the form $f_j(x) = \sum_{i=0}^{k-1} \gamma_i^j x^{2^i}$, $j = 0, 1, 2, \dots, L-1$, where the coefficients γ_i , $i = 0, 1, \dots, k-1$, belong to $GF(2^L)$, L being the L.C.M. of n and k .

We prove that if any of them represents a one-to-one mapping and hence a linear (n, k) code, then the remaining functions also represent linear codes. On the other hand, if any of them represents a many-to-one mapping, then the remaining also represent many-to-one mappings and not linear codes.

Further, if such functions represent linear transformations with respect to a NB of $GF(2^n)$, then they have the feature that they represent transformations with the same *weight distribution*.

We wish to point out that these results are true for any LGSF, whether it be a LF or a LFF. However, we consider the two cases separately in Theorem 5.3.1 and 5.3.2 respectively, for the sake of clarity.

Theorem 5.3.1 states the results for LFs:

Theorem 5.3.1: If a LF, mapping from $GF(2^k)$ to $GF(2^n)$, described by a LP

$f_0(x) = \sum_{i=0}^{k-1} \gamma_i x^{2^i}$, represents a linear (n,k) code, then the LFs described by the LPs,

$$f_j(x) = \sum_{i=0}^{k-1} \gamma_i^{2^j} x^{2^i}, j = 1, 2, \dots, n-1, \quad (5.3.1)$$

also represent linear (n,k) codes, where the coefficients γ_i , $i = 0, 1, \dots, k-1$, belong to $GF(2^n)$.

On the other hand, if $f_0(x)$ does not represent a linear (n,k) code, then $f_j(x)$, $j = 1, 2, \dots, n-1$, also do not represent linear (n,k) codes.

Further, the linear transformations generated by the functions $f_j(x)$, $j = 0, 1, 2, \dots, n-1$, will have the *same weight distribution*, if the vectors in the transformation are considered as elements represented in some *NB* of $GF(2^n)$.

Proof: The coefficients of $f_j(x)$, $j = 0, 1, \dots, n-1$, belong to $GF(2^n)$, since the LGP under consideration is a LP, whose coefficients satisfy trivial conjugacy relations, resulting in the function values and coefficients belonging to the same field $GF(2^n)$.

Now we have

$$f_0(\alpha^u) = \sum_{i=0}^{k-1} \gamma_i \alpha^{u \cdot 2^i}, u = -\infty, 0, 1, \dots, 2^k-2, \quad (5.3.2)$$

where α is a primitive element of $GF(2^k)$, $u \cdot 2^i$ taken modulo 2^k-1 .

We first show that $f_j(.)$'s given by (5.3.1), generate elements which are 2^j th powers of the elements generated by $f_0(x)$, and

$$(f_0(\alpha^u))^{2^j} = f_j(\alpha^{u \cdot 2^j}), \text{ where } u \cdot 2^j \text{ is taken modulo } 2^k-1:$$

Raising (5.3.2) to the power 2^j , we get

$$\begin{aligned} (f_0(\alpha^u))^{2^j} &= \left(\sum_{i=0}^{k-1} \gamma_i \alpha^{u \cdot 2^i} \right)^{2^j} = \sum_{i=0}^{k-1} \gamma_i^{2^j} (\alpha^{u \cdot 2^i})^{2^j}, u = -\infty, 0, 1, \dots, 2^k-2, j = 0, 1, \dots, n-1. \\ &= f_j(\alpha^{u \cdot 2^j}). \end{aligned} \quad (5.3.3)$$

Next we show that the elements generated by the $f_j(.)$'s, $j = 1, 2, \dots, n-1$, are in

fact the code vectors of a linear (n,k) code, iff $f_0(x)$ represents a linear (n,k) code:

Let us assume that $f_0(x)$ represents a linear (n,k) code. Therefore its coefficients satisfy the nonzero determinant property:

$$\det \underline{A} = \det \underline{\Delta}_2 \neq 0.$$

Let the basis vectors of the code generated by $f_0(x)$ be $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$. Now we raise these elements to the power 2^j , and compute the determinant $\underline{\Delta}_2$, with the β_i 's replaced by the respective $\beta_i^{2^j}$'s. Let the new \underline{A} be denoted as \underline{A}' and the new $\underline{\Delta}_2$ by $\underline{\Delta}_2'$. Then

$$\det \underline{A}' = \det \underline{\Delta}_2' = (\det \underline{\Delta}_2)^{2^j}. \quad (5.3.4)$$

This is because $\det \underline{\Delta}_2'$ is a product of 2^j th powers of all the nonzero code vectors generated by $f_0(x)$. Since $\det \underline{\Delta}_2$ is nonzero, $\det \underline{A}'$ given by (5.3.4) is also nonzero, and the corresponding LP represents a linear (n,k) code. Now since the LPs, $f_j(x)$, given by (5.3.1) generate the vectors $\beta_i^{2^j}$, $i = 0, 1, \dots, k-1$, they can form a basis of a code generated by $f_j(x)$. On the other hand, if $f_0(x)$ does not represent a linear (n,k) code, then $\det \underline{\Delta}_2 = 0$ and therefore the remaining $f_j(\cdot)$'s also do not represent linear (n,k) codes, as the corresponding determinant, $\det \underline{\Delta}_2'$, is equal to zero for each $f_j(\cdot)$, $j = 1, 2, \dots, n-1$.

Now we consider the second part of the theorem:

Consider the vectors in the transformations generated by the $f_j(\cdot)$'s. Each f_j generates a linear transformation which contains vectors which are 2^j th powers of the vectors generated by f_0 . Now, if we consider the vectors generated by f_0 as elements in some NB of $GF(2^n)$, then raising them to the power 2^j , results in the cyclic shifting of each of the n -tuple vectors of f_0 by j places, which are present in the transformation generated by f_j . Thus the vectors generated by the $f_j(\cdot)$'s will have the same weight distribution, if they are represented in NB. Q.E.D.

Now we state the results separately for the case of LFFs in Theorem 5.3.2:

Theorem 5.3.2: If a LFF mapping from $GF(2^k)$ to $GF(2^n)$, described by a LFP,

$$f_0(x) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i x^{2^i}) \quad (\text{where } \gamma_i \in GF(2^L), \text{frs}(\Theta) = \Theta + \Theta^Q + \Theta^{Q^2} + \dots + \Theta^{Q^{t-1}}, \Theta^{Q^t} = \Theta,$$

$Q = 2^n$, $L = \text{L.C.M of } n \text{ and } k$, $g = \text{G.C.D of } n \text{ and } k$, and $t = L/n$), represents a linear code, then the LFFs described by the LFPs

$$f_j(x) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i^{2^j} x^{2^i}), \quad j = 1, 2, \dots, L-1, \quad (5.3.5)$$

also represent linear (n, k) codes.

On the other hand, if $f_0(x)$ does not represent a linear (n, k) code, then $f_j(x)$, $j = 1, 2, \dots, L-1$, also do not represent linear (n, k) codes.

Further, the linear transformations generated by the functions $f_j(x)$, $j = 0, 1, 2, \dots, L-1$, will have the *same weight distribution*, if the vectors in the transformation are considered as elements represented in some *NB* of $GF(2^n)$.

Proof: We give proof only for the first part to illustrate the case of LFFs:

$$\text{We have } f_0(\alpha^u) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i \alpha^{u \cdot 2^i}), \quad u = -\infty, 0, 1, \dots, 2^k-2, \quad (5.3.6)$$

where α is a primitive element of $GF(2^k)$, $u \cdot 2^i$ taken modulo 2^k-1 .

We first show that $f_j(\cdot)$'s given by (5.3.5), generate elements which are 2^j th powers of the elements generated by $f_0(x)$, with

$$(f_0(\alpha^u))^{2^j} = f_j(\alpha^{u \cdot 2^j}) \quad \text{where } u \cdot 2^j \text{ is taken modulo } 2^k-1.$$

Raising $f_0(\alpha^u)$ to the power 2^j , we get

$$(f_0(\alpha^u))^{2^j} = \left(\sum_{i=0}^{g-1} \text{frs}(\gamma_i \alpha^{u \cdot 2^i}) \right)^{2^j} = \sum_{i=0}^{g-1} (\text{frs}(\gamma_i \alpha^{u \cdot 2^i}))^{2^j}, \quad u = -\infty, 0, 1, \dots, 2^k-2, \quad j = 0, 1, \dots, L-1.$$

$$= f_j(\alpha^{u \cdot 2^j}),$$

because the i^{th} term in the above expression may be written as

$$\begin{aligned}
 (\text{frs}(\gamma_i \alpha^{u.2^i}))^{2^j} &= ((\gamma_i \alpha^{u.2^i}) + (\gamma_i \alpha^{u.2^i})^Q + \dots + (\gamma_i \alpha^{u.2^i})^{Q^{t-1}})^{2^j} \\
 &= \text{frs}(\gamma_i^{2^j} (\alpha^{u.2^j})^{2^i})
 \end{aligned}$$

Q.E.D.

5.4 Nature of Linear Mappings Generated by Linearized GPs of the form $\beta^j f(x)$, $j = 0, 1, \dots, 2^n - 2$

In Chapter 3, we discussed about the group structure of any GSFs of the form $\beta^j f(x)$, $j = -\infty, 0, 1, \dots, 2^n - 2$. In this section, we are interested in examining the nature of the linear transformations generated by nonzero LGPs which are members of groups of the above form. We show that if a nonzero function $f(x)$, belonging to a group of the above form, represents a one-to-one mapping and thus a linear code, then all the remaining nonzero members of the above group also represent linear codes. On the contrary, if $f(x)$ does not represent a one-to-one mapping, then the remaining nonzero members also do not represent linear codes.

This result is true for any such groups of LGSFs, whether they consist of LFs or LFFs. However, we consider the two cases separately in Theorems 5.4.1 and 5.4.2.

The result in the case of LFs is stated in Theorem 5.4.1:

Theorem 5.4.1: If a LF, mapping from $\text{GF}(2^k)$ to $\text{GF}(2^n)$, described by a LP

$f(x) = \sum_{i=0}^{k-1} \gamma_i x^{2^i}$, represents a linear (n, k) code, then the LFs, described by the LPs

$$\beta^j f(x) = \sum_{i=0}^{k-1} \beta^j \gamma_i x^{2^i}, \quad j = 1, 2, \dots, 2^n - 2, \quad (5.4.1)$$

also represent linear (n, k) codes, where β is a primitive element of $\text{GF}(2^n)$, and the coefficients γ_i , $i = 0, 1, \dots, k-1$, belong to $\text{GF}(2^n)$.

On the other hand, if $f(x)$ represents a many-to-one mapping, then $\beta^j f(x)$, $j = 1, 2, \dots, 2^n - 2$, also represent many-to-one mappings.

Proof: Since we are considering the case of LFs, the conjugacy relations are trivial, thus the coefficients belong to $GF(2^n)$, the same field to which the function values belong.

Now since $f(x)$ represents a linear (n,k) code, we have the relation

$$\det \underline{A} = \det \underline{\Delta}_2 \neq 0.$$

Now we multiply $f(x)$ by β^j which results in all the code vectors being multiplied by β^j . Thus the basis vectors of the code, namely, $\beta_0, \beta_1, \beta_2, \dots, \beta_{k-1}$, change to $\beta^j\beta_0, \beta^j\beta_1, \beta^j\beta_2, \dots, \beta^j\beta_{k-1}$ respectively.

Let the new \underline{A} be denoted as \underline{A}' and the new $\underline{\Delta}_2$ by $\underline{\Delta}_2'$. Then

$$\det \underline{A}' = \det \underline{\Delta}_2' = \beta^{j \cdot 2^k - 1} \det \underline{\Delta}_2. \quad (5.4.2)$$

This is because $\det \underline{\Delta}_2'$ is again a product of all the nonzero ($2^k - 1$ in number) code vectors, which now has β^j as a common factor. Since $f(x)$ represents a linear (n,k) code, $\det \underline{\Delta}_2$ is known to be nonzero. Hence $\det \underline{A}'$ given by (5.4.2) is also nonzero. Thus the LGSFs described by (5.4.1), also represent linear (n,k) codes. On the other hand, if $f(x)$ does not represent a linear code, then $\det \underline{\Delta}_2$ is zero, hence the remaining LPs in the group have their determinants $\det \underline{A}'$ equal to zero, meaning they do not represent one-to-one mappings. Q.E.D.

We restate Theorem 5.4.1 without proof for the case of LFFs in Theorem 5.4.2:

Theorem 5.4.2: If a LFF mapping from $GF(2^k)$ to $GF(2^n)$, described by a LFP

$$f(x) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i x^{2^i}) \quad (\text{where } \gamma_i \in GF(2^L), \text{frs}(\theta) = \theta + \theta^Q + \theta^{Q^2} + \dots + \theta^{Q^{t-1}}, \theta^{Q^t} = \theta,$$

$Q = 2^n$, $L = \text{L.C.M of } n \text{ and } k$, $g = \text{G.C.D. of } n \text{ and } k$, and $t = L/n$), represents a linear (n,k) code, then the LFFs described by the LFPs

$$\beta^j f(x) = \sum_{i=0}^{g-1} \beta^j \text{frs}(\gamma_i x^{2^i}), \quad j = 1, 2, \dots, 2^n - 2, \quad (5.4.3)$$

also represent linear (n,k) codes, where β is a primitive element of $GF(2^n)$.

5.5 Nature of Linear Mappings Generated by Single Term Linearized GPs

In this section, we examine the nature of the linear mappings generated by single term LGPs. In the following theorem, we show that *single term LPs represent only one-to-one mappings*. However, the same cannot be said about single term LFPs.

Theorem 5.5.1: Any single term LP of the form $\beta^j x$, where β is a primitive element of $GF(2^n)$, represents a linear (n,k) code whose $k|n$.

Proof: In (5.2.5c), we put $\gamma_0 = \beta^j$, and $\gamma_i = 0$, $i = 1, 2, \dots, k-1$. This results in the matrix \underline{A} becoming diagonal. Determinant of \underline{A} is then a product of the diagonal elements $\gamma_0^{2^i}$, $i = 0, 1, \dots, k-1$. This determinant is nonzero since γ_0 is nonzero, and $\gamma_0^{2^i}$, $i = 0, 1, \dots, k-1$, belong to a finite field. As $\det \underline{A}$ is nonzero, the mapping described by the single term LP is one-to-one and hence represents a linear (n,k) code. Further, $k|n$, since otherwise we cannot have a single term LP of the form $\beta^j x$ in which the coefficients satisfy nontrivial conjugacy relations. Q.E.D.

Note: Single term LFPs can represent either one-to-one or many-to-one linear mappings, since the matrix \underline{A} is not diagonal in their case (We cannot have a LFP with only the coefficient of x nonzero, and the remaining coefficients zeroes, because of the conjugacy constraints).

5.6 Nature of the Linear Codes Generated by Single Term Linearized GPs which are Members of a Finite Field

We say that two codes are distinct from each other, if there is at least one code vector in one code which is not present in the other. We have seen that the LGP

representation of a linear (n,k) code is not unique, and each linear code can have N_k different representations. Given some LGPs satisfying the nonzero determinant property, we would like to know how many of them represent distinct codes. Grouping LGPs into classes of the form $\beta^j f(x)$, $j = 0, 1, \dots, 2^n - 2$, helps in the study of LGPs representing distinct linear codes. In this section, we conduct such a study of the distinctness of linear codes generated by groups of single term LGPs which have the structure of a finite field isomorphic to $GF(2^n)$. It may be recalled that in Chapter 3, we had discussed about these structures and had denoted the finite fields as F_l and F_f respectively in the case of single term LPs and single term LFPs.

5.6.1 Linear Codes Generated by Single Term LPs of F_l

First, we take up the class of single term LPs which are members of F_l . We examine whether all the linear (n,k) codes generated by the nonzero single term LPs which are members of F_l , are distinct. If all the codes are not distinct, then we compute the number of distinct codes in F_l . We state and prove the relevant results in the following theorem:

Theorem 5.6.1: The number of distinct linear (n,k) codes generated by the nonzero single term LPs of the form $\beta^j x$, $j = 0, 1, 2, \dots, 2^n - 2$, which are members of the finite field F_l , is equal to ν , where $\nu = 2^n - 1 / 2^k - 1$, and the mapping is from $GF(2^k)$ to $GF(2^n)$, $k|n$, with coefficients from $GF(2^n)$, β being a primitive element of $GF(2^n)$. The remaining nonzero members are alternate representations of these ν codes, each code having $2^k - 1$ different representations, thus accounting for all the $2^n - 1$ nonzero members of F_l .

Proof: For proving this, first we note that the nonzero elements of $GF(2^k)$ form a subgroup of the multiplicative group consisting of the nonzero elements of $GF(2^n)$, as $k|n$. Therefore we can form cosets of this subgroup as follows:

1	α	α^2	α^3	.	.	.	α^ξ
β	$\beta\alpha$	$\beta\alpha^2$	$\beta\alpha^3$.	.	.	$\beta\alpha^\xi$
β^2	$\beta^2\alpha$	$\beta^2\alpha^2$	$\beta^2\alpha^3$.	.	.	$\beta^2\alpha^\xi$
.
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$\beta^{\nu-1}$	$\beta^{\nu-1}\alpha$	$\beta^{\nu-1}\alpha^2$	$\beta^{\nu-1}\alpha^3$.	.	.	$\beta^{\nu-1}\alpha^\xi$

where $\nu = 2^n - 1 / 2^k - 1$ and $\xi = 2^k - 2$.

This exhausts all the elements of $GF(2^n)$, and one element appears only once. Further, it may be noted that consecutive values of β^i , $i = 0, 1, \dots, \nu-1$, may be taken as coset leaders, as no β^j appears in any of the cosets corresponding to the coset leaders $\beta^0, \beta, \beta^2, \dots, \beta^{j-1}$, $j < \nu$.

Now we recognize the fact that each row of the above are the nonzero code vectors generated by the single term LPs $\beta^j x$, $j = 0, 1, \dots, \nu-1$. Since the elements in the cosets are distinct, the codes generated by these functions are also distinct.

Now let us consider the set of codes generated by the LPs $\beta^j x$, $j = \nu, \nu+1, \dots, 2\nu-1$. The first function $\beta^\nu x$ generates the code vectors

$$0, \beta^\nu, \beta^\nu \alpha, \beta^\nu \alpha^2, \beta^\nu \alpha^3, \dots, \beta^\nu \alpha^\xi.$$

Since $\beta^\nu = \alpha$, we may express the above code vectors as

$$0, \alpha, \alpha^2, \alpha^3, \alpha^4, \dots, \alpha^\xi, 1.$$

This is same as the code generated by $\beta^0 x$, with the nonzero code vectors cyclically permuted to the left by one place.

Similarly, the second function in this set, namely, $\beta^{\nu+1} x$ generates the code vectors

$$0, \beta^{\nu+1}, \beta^{\nu+1} \alpha, \beta^{\nu+1} \alpha^2, \dots, \beta^{\nu+1} \alpha^\xi \text{ which is equal to}$$

$0, \beta\alpha, \beta\alpha^2, \beta\alpha^3, \dots, \beta$ which is a cyclically permuted version (to the left by one place) of the code generated by βx .

Proceeding in this manner, we may obtain a total of $2^k - 1$ sets, accounting for all the nonzero elements of F_t

Q.E.D.

It follows from the above theorem that any group of functions of the form $f(x) = \beta^i f_0(x)$ where $f_0(x)$ is a LP (not necessarily single term) representing a mapping from $GF(2^k)$ to $GF(2^n)$, $k|n$, which generates the code vectors $0, 1, \alpha, \alpha^2, \dots, \alpha^\xi$; $\xi = 2^k - 2$ (irrespective of the order in which the code vectors are assigned to the message vectors), where α and β are primitive elements of $GF(2^k)$ and $GF(2^n)$ respectively, will have ν distinct codes only, the rest being cyclically permuted versions of these codes. Further, $f_0(x)$ can have coefficients only from the subfield $GF(2^k)$, since the function values belong to $GF(2^k)$ only and hence the mapping is from $GF(2^k)$ to $GF(2^k)$. Since any linear (n, k) code can have N_k different LGP representations, the codes of the above form are distributed in $(N_k/2^k - 1)$ groups, as each group contains $2^k - 1$ different representations of one code.

We cite an example below to illustrate the above results.

Example 5.6.1: Let us take the example of single term LPs generating $(6, 3)$ linear codes. Let $x^6 + x + 1$ be a primitive polynomial for generating $GF(2^6)$. Then the primitive polynomial for generating the subfield $GF(2^3)$ is fixed as $x^3 + x^2 + 1$. Let β and α be primitive elements in $GF(2^6)$ and $GF(2^3)$ respectively. Then $\alpha = \beta^\nu$, where $\nu = 63/7 = 9$. Then the finite field F_7 consisting of single term LPs of the form $\beta^j x$, $j = 0, 1, 2, \dots, 62$, will have 9 distinct codes and each such code will have 7 different representations in F_7 , thus accounting for all the nonzero 63 members of F_7 . We list all the 63 nonzero members of F_7 in Table 5.1. Each member is listed with the coefficients of its LP representation followed by the code vectors generated by it, as a power of β . Only the exponents of β are listed. The table is divided into 7 blocks. The first block contains the 9 distinct codes in this field (the first being $f_0(x)$) and the remaining 6 blocks are cyclic permutations of these codes.

Table 5.1: Nonzero Members of F_7 Comprising of Single Term LPs
Representing Linear (6,3) Codes

$f(x) = \beta^j x$	code vectors as a power of β							
0	β^0	0	9	18	27	36	45	54
1	β^1	1	10	19	28	37	46	55
2	β^2	2	11	20	29	38	47	56
3	β^3	3	12	21	30	39	48	57
4	β^4	4	13	22	31	40	49	58
5	β^5	5	14	23	32	41	50	59
6	β^6	6	15	24	33	42	51	60
7	β^7	7	16	25	34	43	52	61
8	β^8	8	17	26	35	44	53	62
9	β^9	9	18	27	36	45	54	0
10	β^{10}	10	19	28	37	46	55	1
11	β^{11}	11	20	29	38	47	56	2
12	β^{12}	12	21	30	39	48	57	3
13	β^{13}	13	22	31	40	49	58	4
14	β^{14}	14	23	32	41	50	59	5
15	β^{15}	15	24	33	42	51	60	6
16	β^{16}	16	25	34	43	52	61	7
17	β^{17}	17	26	35	44	53	62	8
18	β^{18}	18	27	36	45	54	0	9
19	β^{19}	19	28	37	46	55	1	10
20	β^{20}	20	29	38	47	56	2	11
21	β^{21}	21	30	39	48	57	3	12
22	β^{22}	22	31	40	49	58	4	13
23	β^{23}	23	32	41	50	59	5	14
24	β^{24}	24	33	42	51	60	6	15
25	β^{25}	25	34	43	52	61	7	16
26	β^{26}	26	35	44	53	62	8	17

Table 5.1 (continued)

$f(x) = \beta^j x_j$	code vectors as a power of β							
27	$-\infty$	27	36	45	54	0	9	18
28	$-\infty$	28	37	46	55	1	10	19
29	$-\infty$	29	38	47	56	2	11	20
30	$-\infty$	30	39	48	57	3	12	21
31	$-\infty$	31	40	49	58	4	13	22
32	$-\infty$	32	41	50	59	5	14	23
33	$-\infty$	33	42	51	60	6	15	24
34	$-\infty$	34	43	52	61	7	16	25
35	$-\infty$	35	44	53	62	8	17	26
36	$-\infty$	36	45	54	0	9	18	27
37	$-\infty$	37	46	55	1	10	19	28
38	$-\infty$	38	47	56	2	11	20	29
39	$-\infty$	39	48	57	3	12	21	30
40	$-\infty$	40	49	58	4	13	22	31
41	$-\infty$	41	50	59	5	14	23	32
42	$-\infty$	42	51	60	6	15	24	33
43	$-\infty$	43	52	61	7	16	25	34
44	$-\infty$	44	53	62	8	17	26	35
45	$-\infty$	45	54	0	9	18	27	36
46	$-\infty$	46	55	1	10	19	28	37
47	$-\infty$	47	56	2	11	20	29	38
48	$-\infty$	48	57	3	12	21	30	39
49	$-\infty$	49	58	4	13	22	31	40
50	$-\infty$	50	59	5	14	23	32	41
51	$-\infty$	51	60	6	15	24	33	42
52	$-\infty$	52	61	7	16	25	34	43
53	$-\infty$	53	62	8	17	26	35	44

Theorem 5.6.2: The number of distinct linear (n,k) codes generated by the nonzero single term LFPs of the form $\beta^i \text{frs}(\gamma^j x)$, $i = 0, 1, 2, \dots, 2^n - 2$, which are members of the finite field F_f , is equal to ν_2 , where $\nu_2 = 2^n - 1 / 2^g - 1$, and the mapping is from $GF(2^k)$ to $GF(2^n)$, $k \nmid n$, with coefficients from $GF(2^L)$, $L = \text{L.C.M of } n \text{ and } k$, β being a primitive element of $GF(2^n)$. The remaining nonzero members are alternate representations of these ν_2 codes, each code having $2^g - 1$ different representations, thus accounting for all the $2^n - 1$ nonzero members of F_f

Proof: We first note that, by multiplying a sequence belonging to $GF(2^n)$ of length $2^k - 1$, generated by single term LFPs, by β^i , $i = 1, 2, \dots, 2^n - 2$, β being a primitive element of $GF(2^n)$, the only possible permutations of the sequence are cyclic in nature. In the case of single term LPs, we had a similar situation, where the nonzero code vectors generated by $f(x) = \beta^i x$, $i = 0, 1, \dots, \nu - 1$, were cyclically shifted to the left by one place when multiplied by β^ν , $\nu = 2^n - 1 / 2^k - 1$. We would like to generalize this result to any n and k , where k does not necessarily divide n . To start with, we take the identity element of F_f , i.e., $\text{frs}(\gamma^j x)$ where $\text{frs}(\gamma^j) = 1$, analogous to $f_0(x) = x$, in the case where $k \mid n$. The function $f_0(x) = x$ generated all the elements of $GF(2^k)$, since $GF(2^k)$ is a subfield of $GF(2^n)$. However, when $k \nmid n$, the corresponding function $\text{frs}(\gamma^j x)$ cannot generate all the elements of $GF(2^k)$, since $GF(2^k)$ is not a subfield of $GF(2^n)$ in this case. However, it does generate all the elements of subfields which are common to both $GF(2^k)$ and $GF(2^n)$, or in other words, all the elements of subfields whose extension order divides both k and n , which has a maximum value equal to the G.C.D. of n and k , say g . Let us find the values of x at which the function $\text{frs}(\gamma^j x)$ assumes values from $GF(2^g)$. Let δ be a primitive element of $GF(2^g)$ and let α be a primitive element of $GF(2^k)$. Then $\delta = \alpha^{\nu_1} = \beta^{\nu_2}$, where $\nu_1 = 2^k - 1 / 2^g - 1$ and $\nu_2 = 2^n - 1 / 2^g - 1$. Now as $\text{frs}(\gamma^j) = 1$, multiplying both sides by δ^i , we get

$$\delta^i \text{frs}(\gamma^j) = \delta^i, i = 0, 1, \dots, 2^g - 2.$$

Taking δ^i inside the argument, we get

Theorem 5.6.2: The number of distinct linear (n,k) codes generated by the nonzero single term LFPs of the form $\beta^i \text{frs}(\gamma^j x)$, $i = 0, 1, 2, \dots, 2^n - 2$, which are members of the finite field F_f , is equal to ν_2 , where $\nu_2 = 2^n - 1 / 2^g - 1$, and the mapping is from $GF(2^k)$ to $GF(2^n)$, $k \nmid n$, with coefficients from $GF(2^L)$, $L = \text{L.C.M of } n \text{ and } k$, β being a primitive element of $GF(2^n)$. The remaining nonzero members are alternate representations of these ν_2 codes, each code having $2^g - 1$ different representations, thus accounting for all the $2^n - 1$ nonzero members of F_f

Proof: We first note that, by multiplying a sequence belonging to $GF(2^n)$ of length $2^k - 1$, generated by single term LFPs, by β^i , $i = 1, 2, \dots, 2^n - 2$, β being a primitive element of $GF(2^n)$, the only possible permutations of the sequence are cyclic in nature. In the case of single term LPs, we had a similar situation, where the nonzero code vectors generated by $f(x) = \beta^i x$, $i = 0, 1, \dots, \nu - 1$, were cyclically shifted to the left by one place when multiplied by β^ν , $\nu = 2^n - 1 / 2^k - 1$. We would like to generalize this result to any n and k , where k does not necessarily divide n . To start with, we take the identity element of F_f , ie., $\text{frs}(\gamma^j x)$ where $\text{frs}(\gamma^j) = 1$, analogous to $f_0(x) = x$, in the case where $k \mid n$. The function $f_0(x) = x$ generated all the elements of $GF(2^k)$, since $GF(2^k)$ is a subfield of $GF(2^n)$. However, when $k \nmid n$, the corresponding function $\text{frs}(\gamma^j x)$ cannot generate all the elements of $GF(2^k)$, since $GF(2^k)$ is not a subfield of $GF(2^n)$ in this case. However, it does generate all the elements of subfields which are common to both $GF(2^k)$ and $GF(2^n)$, or in other words, all the elements of subfields whose extension order divides both k and n , which has a maximum value equal to the G.C.D. of n and k , say g . Let us find the values of x at which the function $\text{frs}(\gamma^j x)$ assumes values from $GF(2^g)$. Let δ be a primitive element of $GF(2^g)$ and let α be a primitive element of $GF(2^k)$. Then $\delta = \alpha^{\nu_1} = \beta^{\nu_2}$, where $\nu_1 = 2^k - 1 / 2^g - 1$ and $\nu_2 = 2^n - 1 / 2^g - 1$. Now as $\text{frs}(\gamma^j) = 1$, multiplying both sides by δ^i , we get

$$\delta^i \text{frs}(\gamma^j) = \delta^i, i = 0, 1, \dots, 2^g - 2.$$

Taking δ^i inside the argument, we get

$$\text{frs}(\gamma^j \delta^i) = \delta^i, i = 0, 1, \dots, 2^g-2. \quad (5.6.1)$$

ie., at $x = \delta^i = (\alpha^{\nu_1})^i, i = 0, 1, \dots, 2^g-2$, the function assumes values of δ^i .

Thus the code vectors generated by $\text{frs}(\gamma^j x)$ are as follows:

$$\begin{aligned} &0, \text{frs}(\gamma^j), \text{frs}(\gamma^j \alpha), \text{frs}(\gamma^j \alpha^2), \dots, \text{frs}(\gamma^j \alpha^{\nu_1-1}), \\ &\text{frs}(\gamma^j \delta), \text{frs}(\gamma^j \delta \alpha), \text{frs}(\gamma^j \delta \alpha^2), \dots, \text{frs}(\gamma^j \delta \alpha^{\nu_1-1}), \\ &\text{frs}(\gamma^j \delta^2), \dots, \text{frs}(\gamma^j \delta^2 \alpha^{\nu_1-1}), \dots, \\ &\text{frs}(\gamma^j \delta^{2^g-2}), \text{frs}(\gamma^j \delta^{2^g-2} \alpha), \text{frs}(\gamma^j \delta^{2^g-2} \alpha^2), \dots, \text{frs}(\gamma^j \delta^{2^g-2} \alpha^{\nu_1-1}). \end{aligned}$$

Now we multiply $\text{frs}(\gamma^j x)$ by $\delta = \beta^{\nu_2}$. After taking $\delta = \beta^{\nu_2}$ inside the argument of $\text{frs}(\gamma^j x)$, this gives the following sequence:

$$\begin{aligned} &0, \text{frs}(\gamma^j \delta), \text{frs}(\gamma^j \delta \alpha), \text{frs}(\gamma^j \delta \alpha^2), \dots, \text{frs}(\gamma^j \delta \alpha^{\nu_1-1}), \\ &\text{frs}(\gamma^j \delta^2), \dots, \text{frs}(\gamma^j \delta^2 \alpha^{\nu_1-1}), \dots, \\ &\text{frs}(\gamma^j \delta^{2^g-2}), \text{frs}(\gamma^j \delta^{2^g-2} \alpha), \text{frs}(\gamma^j \delta^{2^g-2} \alpha^2), \dots, \text{frs}(\gamma^j \delta^{2^g-2} \alpha^{\nu_1-1}), \\ &\text{frs}(\gamma^j), \text{frs}(\gamma^j \alpha), \text{frs}(\gamma^j \alpha^2), \dots, \text{frs}(\gamma^j \alpha^{\nu_1-1}). \end{aligned}$$

This same as the sequence generated by $\text{frs}(\gamma^j x)$ except that its nonzero values are cyclically permuted to the left by $\nu_1 = 2^k-1/2^g-1$ places. In general, multiplying $\text{frs}(\gamma^j x)$ by $\delta^i, i = 0, 1, \dots, 2^g-2$, cyclically permutes the sequence generated by it, by $(2^k-1/2^g-1).i$ places. Thus the number of distinct linear codes in F_f must be ν_2 , and the remaining linear codes in F_f are cyclic permutations of these ν_2 codes, the number of different representations of each distinct code present in F_f being 2^g-1 . Q.E.D.

Note: We see that, when $g = 1$, F_f contains only one representation of each code. Thus all the 2^n-1 codes in F_f are distinct when n and k are relatively prime.

It follows that any group of functions of the form $f(x) = \beta^j f_0(x)$ where

$$f_0(x) = \sum_{i=0}^{g-1} \text{frs}(\gamma_i x^{2^i}), j = -\infty, 0, 1, \dots, 2^n-2, \text{ (where } \gamma_i \in \text{GF}(2^L), \text{frs}(\Theta) = \Theta + \Theta^Q + \Theta^{Q^2} + \dots + \Theta^{Q^{t-1}}, \Theta^{Q^t} = \Theta, Q = 2^n, L = \text{L.C.M of } n \text{ and } k, g = \text{G.C.D. of } n \text{ and } k, \text{ and } t = L/n,$$
 β being a primitive element of $\text{GF}(2^n)$), (ie., $f_0(x)$ is not necessarily single term) which generates the code vectors generated by the single term LFP $\text{frs}(\gamma^j x)$ (irrespective of the order in which the code vectors are assigned to the message vectors), will have ν_2 distinct codes only, the rest being cyclically permuted versions of these codes. Since any linear (n,k) code of this form can have N_k different LFP representations, the codes of the above form are distributed in $(N_k/2^g-1)$ groups, as each group contains 2^g-1 different representations of one code.

We give some examples of F_f to illustrate the above results:

Example 5.6.2: Let $n = 3, k = 2$. Here the G.C.D. of n and $k = g = 1$. Hence all the codes generated by the nonzero single term LFPs in F_f are distinct. This is true as the number of distinct linear $(3,2)$ codes $= 42/6 = 7$, and the number of nonzero LFPs in F_f is also equal to 7. Each code has $N_k = 6$ different representations and these are distributed in 6 isomorphic finite fields, thus accounting for all the 42 linear $(3,2)$ transformations representing linear $(3,2)$ codes. The coefficients of these polynomials belong to $\text{GF}(2^6)$. Let $x^6 + x + 1$ be a primitive polynomial for generating this field, with γ as a primitive element. Then the fields $\text{GF}(2^3)$ and $\text{GF}(2^2)$ are generated by the primitive polynomials $x^3 + x^2 + 1$ and $x^2 + x + 1$ respectively, with primitive elements β and α respectively.

We first choose those elements γ^j of $\text{GF}(2^6)$ whose Frobenius sum with respect to $\text{GF}(2^3)$ is 1, ie., $\sum_{m=0}^1 \gamma^{jQ^m} = 1$, and whose corresponding LFP $\text{frs}(\gamma^j x)$ generates a valid linear $(3,2)$ code. These serve as identities in the respective fields. These values were earlier listed in Example 3.9.1 as $\gamma^j, j = 11, 22, 25, 37, 44$ and 50. The nonzero functions in each

field represent linear (3,2) codes. The 6 different representations of each of the 7 distinct codes are distributed in 6 fields, as listed in Table 5.3. Each member of a field is listed with its LFP representation $\beta^j \text{frs}(\gamma^j x) = \text{frs}(\gamma^{j+9i} x)$, $i = 0, 1, \dots, 2^n - 2$, followed by the code vectors generated by it as a power of β . Only the exponents of β and γ are listed (The trivial function 0 is not listed).

Table 5.3: Single Term Linearized Frobenius Polynomials Representing One-to-One Linear (3,2) Transformations Grouped into 6 Isomorphic Finite Fields

(1)

$\text{frs}(\gamma^{j+9i} x)$ $j + 9i$	code vectors as a power of β			
11	$-\infty$	0	5	1
20	$-\infty$	1	6	2
29	$-\infty$	2	0	3
38	$-\infty$	3	1	4
47	$-\infty$	4	2	5
56	$-\infty$	5	3	6
2	$-\infty$	6	4	0

(2)

$\text{frs}(\gamma^{j+9i} x)$ $j + 9i$	code vectors as a power of β			
22	$-\infty$	0	2	3
31	$-\infty$	1	3	4
40	$-\infty$	2	4	5
49	$-\infty$	3	5	6
58	$-\infty$	4	6	0
4	$-\infty$	5	0	1
13	$-\infty$	6	1	2

Table 5.3 (continued)

(3)

$\text{frs}(\gamma^{j+9.i}x)$ $j + 9.i$	code vectors as a power of β			
25	$-\infty$	0	1	5
34	$-\infty$	1	2	6
43	$-\infty$	2	3	0
52	$-\infty$	3	4	1
61	$-\infty$	4	5	2
7	$-\infty$	5	6	3
16	$-\infty$	6	0	4

(4)

$\text{frs}(\gamma^{j+9.i}x)$ $j + 9.i$	code vectors as a power of β			
37	$-\infty$	0	4	6
46	$-\infty$	1	5	0
55	$-\infty$	2	6	1
1	$-\infty$	3	0	2
10	$-\infty$	4	1	3
19	$-\infty$	5	2	4
28	$-\infty$	6	3	5

(5)

$\text{frs}(\gamma^{j+9.i}x)$ $j + 9.i$	code vectors as a power of β			
44	$-\infty$	0	6	4
53	$-\infty$	1	0	5
62	$-\infty$	2	1	6
8	$-\infty$	3	2	0
17	$-\infty$	4	3	1
26	$-\infty$	5	4	2
35	$-\infty$	6	5	3

Table 5.3 (continued)

(6)

$\text{frs}(\gamma^{j+9.i}x)$ $j + 9.i$	code vectors as a power of β			
50	$-\infty$	0	3	2
59	$-\infty$	1	4	3
5	$-\infty$	2	5	4
14	$-\infty$	3	6	5
23	$-\infty$	4	0	6
32	$-\infty$	5	1	0
41	$-\infty$	6	2	1

Example 5.6.3: Let $n = 6$, $k = 4$. Thus $g = 2$. Any $(6,4)$ linear code can be represented, in general, by LFPs consisting of two linearized Frobenius terms, out of which we select only the single term LFPs. According to Theorem 5.6.2, $\nu_2 = 2^6 - 1/2^2 - 1 = 21$ codes generated by the nonzero single term LFPs of F_f should be distinct, and there should be $2^g - 1 = 3$ different representations of each distinct code present in F_f each of whose nonzero code vectors being cyclically permuted versions of the first (to the left) by $(2^k - 1/2^g - 1).i = 5.i$ places ($i = 0, 1, 2$). Each code has $N_k = 15 \times 14 \times 12 \times 8 = 20160$ different representations and these are distributed in $N_k/2^g - 1 = 20160/3 = 6720$ groups, including F_f . Each such group consists of 3 blocks, each block containing 21 codes, the codes in the second and third block being alternate representations of the codes in the first block. Listed in Table 5.4, are the elements of the first block of F_f containing 21 codes.

The coefficients of these LFPs belong to $\text{GF}(2^{12})$. Let $x^{12} + x^6 + x^4 + x + 1$ be a primitive polynomial for generating this field, with γ as a primitive element. Then the

Table 5.4: First Block of Nonzero Members of F_f comprising of Single Term Linearized Frobenius Polynomials Representing Distinct Linear (6,4) Codes

$\text{frs}(\gamma^u x)$ $u=j+65.i$	Code vectors as a power of β															
197	$-\infty$	0	51	5	48	61	21	9	26	6	19	42	30	47	27	40
262	$-\infty$	1	52	6	49	62	22	10	27	7	20	43	31	48	28	41
327	$-\infty$	2	53	7	50	0	23	11	28	8	21	44	32	49	29	42
392	$-\infty$	3	54	8	51	1	24	12	29	9	22	45	33	50	30	43
457	$-\infty$	4	55	9	52	2	25	13	30	10	23	46	34	51	31	44
522	$-\infty$	5	56	10	53	3	26	14	31	11	24	47	35	52	32	45
587	$-\infty$	6	57	11	54	4	27	15	32	12	25	48	36	53	33	46
652	$-\infty$	7	58	12	55	5	28	16	33	13	26	49	37	54	34	47
717	$-\infty$	8	59	13	56	6	29	17	34	14	27	50	38	55	35	48
782	$-\infty$	9	60	14	57	7	30	18	35	15	28	51	39	56	36	49
847	$-\infty$	10	61	15	58	8	31	19	36	16	29	52	40	57	37	50
912	$-\infty$	11	62	16	59	9	32	20	37	17	30	53	41	58	38	51
977	$-\infty$	12	0	17	60	10	33	21	38	18	31	54	42	59	39	52
1042	$-\infty$	13	1	18	61	11	34	22	39	19	32	55	43	60	40	53
1107	$-\infty$	14	2	19	62	12	35	23	40	20	33	56	44	61	41	54
1172	$-\infty$	15	3	20	0	13	36	24	41	21	34	57	45	62	42	55
1237	$-\infty$	16	4	21	1	14	37	25	42	22	35	58	46	0	43	56
1302	$-\infty$	17	5	22	2	15	38	26	43	23	36	59	47	1	44	57
1367	$-\infty$	18	6	23	3	16	39	27	44	24	37	60	48	2	45	58
1432	$-\infty$	19	7	24	4	17	40	28	45	25	38	61	49	3	46	59
1497	$-\infty$	20	8	25	5	18	41	29	46	26	39	62	50	4	47	60

fields $GF(2^6)$ and $GF(2^4)$ are generated by the primitive polynomials $x^6 + x^5 + 1$ and $x^4 + x + 1$ respectively, with primitive elements $\beta = \gamma^{65}$ and $\alpha = \gamma^{273}$ respectively. Only the exponents of γ and β are listed in the table. In the table, the first function is, say, $\text{frs}(\gamma^j x)$ (the identity element of F_f) and the remaining functions are of the form, $\text{frs}(\gamma^u x)$, where $u = j + 65i$, $i = 0, 1, 2, \dots, 20$.

The first code in the second block is then generated by $\text{frs}(\gamma^{1562} x)$ as β^i , $i = -\infty, 21, 9, 26, 6, 19, 42, 30, 47, 27, 40, 0, 51, 5, 48, 61$, which is the same as the code generated by $\text{frs}(\gamma^{197} x)$, the first code in the first block, except that the nonzero code vectors in the former are a cyclically permuted (to the left by 5 places) version of the codes in the latter. Similarly, the second code in the second block would be generated by $\text{frs}(\gamma^{1627} x)$ as β^i , $i = -\infty, 22, 10, 27, 7, 20, 43, 31, 48, 28, 41, 1, 52, 6, 49, 62$, which is the same as the second code in the first block, except that the nonzero code vectors in the former are cyclically permuted to the left by 5 places. Thus all the codes in the second block are same as the codes in the first block, except that the nonzero code vectors are cyclically permuted to the left by 5 places. Similarly, it may be seen that the codes in the third block are cyclically permuted versions (to the left by 10 places) of the codes in the first block.

Now we list in Table 5.5, the first block of one of the 6720 groups other than F_f whose LFP representations consist of two terms, but generate the same group of codes as in F_f , with the code vectors permuted in some noncyclic manner with respect to that of F_f . In the table, the first function is, say, $\text{frs}(\gamma^{j_1} x) + \text{frs}(\gamma^{j_2} x^2)$, and the remaining functions are of the form, $\text{frs}(\gamma^u x) + \text{frs}(\gamma^v x^2)$, where $u = j_1 + 65i$ and $v = j_2 + 65i$, $i = 0, 1, 2, \dots, 20$.

Table 5.5: First Block of Nonzero Members of a Group Comprising of Multiple Term Linearized Frobenius Polynomials Representing the Same Set of Distinct Linear Codes as in Table 5.4

$\text{frs}(\gamma^u x)$ $+$ $\text{frs}(\gamma^v x^2)$		Code vectors as a power of β															
u	v																
905	0	$-\infty$	0	5	51	48	6	21	19	27	61	9	42	30	47	26	40
970	65	$-\infty$	1	6	52	49	7	22	20	28	62	10	43	31	48	27	41
1035	130	$-\infty$	2	7	53	50	8	23	21	29	0	11	44	32	49	28	42
1100	195	$-\infty$	3	8	54	51	9	24	22	30	1	12	45	33	50	29	43
1165	260	$-\infty$	4	9	55	52	10	25	23	31	2	13	46	34	51	30	44
1230	325	$-\infty$	5	10	56	53	11	26	24	32	3	14	47	35	52	31	45
1295	390	$-\infty$	6	11	57	54	12	27	25	33	4	15	48	36	53	32	46
1360	455	$-\infty$	7	12	58	55	13	28	26	34	5	16	49	37	54	33	47
1425	520	$-\infty$	8	13	59	56	14	29	27	35	6	17	50	38	55	34	48
1490	585	$-\infty$	9	14	60	5	15	30	28	36	7	18	51	39	56	35	49
1555	650	$-\infty$	10	15	61	58	16	31	29	37	8	19	52	40	57	36	50
1620	715	$-\infty$	11	16	62	59	17	32	30	38	9	20	53	41	58	37	51
1685	780	$-\infty$	12	17	0	60	18	33	31	39	10	21	54	42	59	38	52
1750	845	$-\infty$	13	18	1	61	19	34	32	40	11	22	55	43	60	39	53
1815	910	$-\infty$	14	19	2	62	20	35	33	41	12	23	56	44	61	40	54
1880	975	$-\infty$	15	20	3	0	21	36	34	42	13	24	57	45	62	41	55
1945	1040	$-\infty$	16	21	4	1	22	37	35	43	14	25	58	46	0	42	56
2010	1105	$-\infty$	17	22	5	2	23	38	36	44	15	26	59	47	1	43	57
2075	1170	$-\infty$	18	23	6	3	24	39	37	45	16	27	60	48	2	44	58
2140	1235	$-\infty$	19	24	7	4	25	40	38	46	17	28	61	49	3	45	59
2205	1300	$-\infty$	20	25	8	5	26	41	39	47	18	29	62	50	4	46	60

5.7 Roots of Linearized GPs Representing Linear Codes

In this section, we discuss about the roots of LGPs representing linear codes. It is known that the roots of a LGP form a subspace (for theory of LGPs, see Appendix A). The roots may belong to $GF(2^L)$, or in some extension of it.

A question which naturally arises is whether one can characterize linear codes by the roots of their LGP representations. The answer is that the roots do not characterize individual codes, as we saw that multiplication of a LGP representing a linear (n,k) code, by β^j , where β is a primitive element of $GF(2^n)$, in general, gives rise to another linear code which is distinct from the first. However, both the LGPs representing distinct codes would have the same set of roots, as the second polynomial is obtained by multiplying the first by a constant. It follows that all the LGPs in a group of the form $\beta^j f(x)$, $j = 0, 1, \dots, 2^n - 2$, have the same set of roots. Each such group has a distinct set of roots. Hence one can say that, *the roots of LGPs characterize a group of codes rather than individual codes.*

We observe one constraint on these set of roots of any LGP representing a linear (n,k) code. That is, *the roots cannot assume nonzero values from $GF(2^k)$.* This is so, because, if the roots assumed values from $GF(2^k)$, then this would mean that the LGP assumes a function value of 0 at an input other than $x = 0$. In other words, this would mean that there is a code vector 0, assigned to a nonzero message vector, which is not true. Hence the root space of any LGP representing a linear (n,k) code cannot assume nonzero values from $GF(2^k)$.

5.8 Representation of Cyclic Codes by Linearized GPs

In this section, we find representations of cyclic codes in terms of LGPs, both in SB and in NB.

We compute the LGP coefficients in the respective basis as follows:

Let α and β be primitive elements of $GF(2^k)$ and $GF(2^n)$ in SB respectively, and let δ and σ be primitive elements of $GF(2^k)$ and $GF(2^n)$ in NB respectively. Then a SB LGP is

one whose coefficients are computed with the code vectors (n -tuples) considered as powers of β and the message vectors (k -tuples) considered as powers of α . Similarly, a *NB LGP* is one whose coefficients are computed with the same code vectors and message vectors now considered as powers of σ and δ respectively.

Every cyclic code can be generated by a polynomial $g(x)$ of degree $n-k$, called the generator polynomial. We may represent $g(x)$ as a polynomial in β , of the form

$$g(\beta) = a_{n-k}\beta^{n-k} + a_{n-k-1}\beta^{n-k-1} + \dots + a_1\beta + a_0\beta^0, \quad (5.8.1)$$

where a_i 's $\in GF(2)$ and β is a primitive element in $GF(2^n)$.

Now the LGP representation of any linear (n,k) code depends on the basis vectors $f(\alpha^i)$, $i = 0, 1, \dots, k-1$, chosen for the code. Hence, for representing a (n,k) cyclic code by a LGP, we choose one of the bases of the code as the canonic basis and derive its LGP representation. The basis chosen for the cyclic code is the one in which (1) the bottom row of its generator matrix \underline{G} , has the rightmost $n-k+1$ entries ($\in GF(2)$) as the coefficients a_i , $i = 0, 1, \dots, n-k$, of the generator polynomial $g(x)$, with the leftmost $k-1$ entries being 0, (2) the next to bottom row consisting of the coefficients of $x.g(x)$ modulo x^n-1 , which is the same as the bottom row cyclically shifted to the left by one place, and so on. i.e., \underline{G} is an $k \times n$ matrix, of the form

$$\underline{G} = \begin{bmatrix} a_{n-k} & a_{n-k-1} & & a_0 & 0 & \dots & 0 & 0 \\ 0 & a_{n-k} & a_{n-k-1} & \dots & a_0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & 0 & 0 \\ 0 & 0 & \dots & a_{n-k} & a_{n-k-1} & \dots & a_0 & 0 \\ 0 & 0 & \dots & 0 & a_{n-k} & a_{n-k-1} & \dots & a_0 \end{bmatrix} \quad (5.8.2)$$

With the above canonic basis, we find suitable representations for a cyclic code both in SB and in NB, in the next two subsections:

5.8.1 Standard Basis Representation

Theorem 5.8.1: The LGP coefficients of a linear (n,k) cyclic code can be expressed in SB as

$$\underline{A} = \beta^i \underline{V}_s^{-1} \underline{B}, \quad (5.8.3)$$

if the basis chosen for the code is the canonic basis given in (5.8.2) and where β^i is the generator polynomial of the cyclic code represented as a power of a primitive element β in $GF(2^n)$, \underline{A} is the coefficient vector in SB, given by

$\begin{bmatrix} a_{-2^0} & a_{-2^1} & \dots & a_{-2^i} & \dots & a_{-2^{k-1}} \end{bmatrix}^T$ (where -2^i is taken modulo 2^k-1), $\underline{B} = [\beta^0 \ \beta \ \beta^2 \ \beta^3 \ \dots \ \beta^{k-1}]^T$ consists of the 'k' SB vectors of $GF(2^n)$, T denotes transpose, \underline{V}_s^{-1} is the inverse of the Vander monde matrix as given by (3.7.3) and SB is used for representing the elements of $GF(2^k)$.

Proof: Each row of the generator matrix, consisting of n elements from $GF(2)$, can also be considered as a single element belonging to the extension field $GF(2^n)$. Now, in the canonic basis chosen for the cyclic code, the bottom most row has the coefficients of the generator polynomial $g(x)$ of the code. Since the degree of this polynomial is $n-k$, the leftmost $k-1$ elements in this row will be zero. This polynomial can be considered as an element belonging to the extension field $GF(2^n)$, can be denoted as a power of a primitive element β in $GF(2^n)$, say β^i , and can be expressed in terms of the SB vectors of $GF(2^n)$, $\beta^0, \beta, \beta^2, \beta^3, \dots, \beta^{n-1}$, as

$$\beta^i = 0. \beta^{n-1} + \dots + 0. \beta^{n-k+1} + a_{n-k} \beta^{n-k} + a_{n-k-1} \beta^{n-k-1} + \dots + a_1 \beta + a_0 \beta^0.$$

The next to bottom row consists of the coefficients of $x.g(x)$ modulo x^n-1 , which is the same as the bottom row cyclically shifted to the left by one place. We can see that, this row expressed as a power of β is β^{i+1} . It may be further noted that, the row corresponding to $x^2.g(x)$ modulo x^n-1 may be expressed as β^{i+2} , and so on. Finally the row corresponding to $x^{k-1}.g(x)$ modulo x^n-1 may be expressed as β^{i+k-1} . This is because $g(x) = \beta^i$ has $k-1$

leftmost positions equal to zeroes, and hence multiplication of $g(x)$ by x^j modulo x^n-1 has the same effect as multiplication of β^j by β^j modulo the irreducible polynomial for generating $GF(2^n)$, $j = 1, 2, \dots, k-1$, since the modulo polynomial of $GF(2^n)$ does not come into the picture for $1 \leq j \leq k-1$. Therefore in the relation $\underline{A} = \underline{V}_s^{-1} \underline{f}$, we substitute \underline{f}

$$= \begin{bmatrix} f(\alpha^0) & f(\alpha) & f(\alpha^2) & \dots & f(\alpha^{k-1}) \end{bmatrix}^T \text{ as } \underline{f} = \begin{bmatrix} \beta^j, \beta^{j+1}, \beta^{j+2}, \beta^{j+3}, \dots, \beta^{j+k-1} \end{bmatrix}^T$$

$$= \beta^j \begin{bmatrix} \beta^0, \beta^1, \beta^2, \beta^3, \dots, \beta^{k-1} \end{bmatrix}^T = \beta^j \underline{B}.$$

$$\text{Hence } \underline{A} = \beta^j \underline{V}_s^{-1} \underline{B}.$$

Q.E.D.

5.8.2 Normal Basis Representation

Theorem 5.8.2: The LGP coefficients of a linear (n,k) cyclic code can be expressed in NB as

$$\underline{A} = \underline{V}_n^{-1} \underline{f}_c, \quad (5.8.4)$$

if the basis chosen for the code is the canonic basis given in (5.8.2), \underline{V}_n^{-1} is the inverse of the Vander monde matrix as given in (3.7.9) when NB is used for representing the elements of $GF(2^k)$, \underline{A} is the coefficient vector in NB given by

$$\begin{bmatrix} a_{-2^0} & a_{-2^1} & \dots & a_{-2^i} & \dots & a_{-2^{k-1}} \end{bmatrix}^T \text{ (where } -2^i \text{ is taken modulo } 2^k-1), \underline{f}_c = \begin{bmatrix} y, y^2, y^{2^2}, \dots, y^{2^{k-1}} \end{bmatrix}^T$$

, where y is same as the generator polynomial $g(x) = \beta^j$, but is now recognized as an element of $GF(2^n)$ represented in NB.

Proof: We know that the LGP coefficients of any linear (n,k) code in NB is given by

$\underline{A} = \underline{V}_n^{-1} \underline{f}$, where $\underline{f} = \begin{bmatrix} f(\delta^0) & f(\delta) & f(\delta^2) & \dots & f(\delta^{k-1}) \end{bmatrix}^T$, δ being a primitive element in $GF(2^k)$. Now, in the canonic basis chosen for the cyclic code, the bottom most row is $g(x) = f(\delta^0)$ equal to y which is considered as an element of $GF(2^n)$ represented in NB. The next to bottom row consists of the coefficients of $x.g(x)$ modulo x^n-1 , which is the same as

the bottom row cyclically shifted to the left by one place. We may note that a cyclic shift of the components of an element represented in NB results in squaring of that element. Thus the next to bottom row, ie., $f(\delta)$ can be written in terms of y as y^2 , $f(\delta^2)$ may be written as y^{2^2} , and so on. Finally the row corresponding to $x^{k-1} \cdot g(x)$ modulo $x^n - 1$ may be expressed as $y^{2^{k-1}}$. Q.E.D.

Now we state and prove a theorem on some (n,k) cyclic codes whose $k|n$ and which has a generator polynomial y which can be recognized as an element in $GF(2^n)$ as well as in the subfield $GF(2^k)$, in NB, ie., $y^{2^k} = y$. In fact the theorem is valid if the cyclic code contains any y , not necessarily the generator polynomial, such that $y^{2^k} = y$, and $y, y^2, y^{2^2}, \dots, y^{2^{k-1}}$ are linearly independent.

Theorem 5.8.3: If there exists an (n,k) cyclic code, whose $k|n$, which has a generator polynomial y , which when considered as an element of $GF(2^n)$ in NB, has the property that $y^{2^k} = y$, ie., y also belongs to the subfield $GF(2^k)$, then the code can be represented in NB

(1) by a LP with coefficients from the ground field, ie., by a p -polynomial, if the canonic basis is used for the code

(2) by a LP with coefficients from $GF(2^k)$, in general, if any other basis is used for the code.

Note: If there exists a y such that $y^{2^k} = y$, then k is the least positive integer with which this is true. For it were not so, then it means that the degree of the generator polynomial is greater than $n-k$, which is not true.

Proof: For proving (1), we use the relation $A = V_n^{-1} f_c$ derived in Theorem 5.8.2, and apply the condition that $y^{2^k} = y$. Then the coefficients can be expressed as a trace function as

$$a_{-2^i} = \text{tr}(b_0^{2^i} y), \quad i = 0, 1, \dots, k-1, \quad (5.8.5)$$

(where $\text{tr}(\Theta) = \Theta + \Theta^2 + \Theta^4 + \dots + \Theta^{2^{k-1}}$, $\Theta \in \text{GF}(2^k)$, and -2^i taken modulo $2^k - 1$), as is evident from the matrix relation, remembering that b_0 and y both now belong to $\text{GF}(2^k)$. Since trace always belongs to the ground field, (in this case, $\text{GF}(2)$), the coefficients are either 0 or 1. Hence the code has a p-polynomial representation.

For proving (2), we take note of the fact that since the code already contains linearly independent vectors of the form $y, y^2, y^{2^2}, \dots, y^{2^{k-1}}$, with $y^{2^k} = y$, y thus belonging to $\text{GF}(2^k)$, a linear combination of these vectors also belongs to $\text{GF}(2^k)$. Hence the code contains code vectors all of them belonging to $\text{GF}(2^k)$ in NB. Thus we have a mapping from $\text{GF}(2^k)$ to $\text{GF}(2^k)$, resulting in the LP coefficients also belonging to $\text{GF}(2^k)$.

If $y \in \text{GF}(2^n)$ also belongs to $\text{GF}(2^k)$, then $\text{GF}(2^k)$ should be a subfield of $\text{GF}(2^n)$. Therefore k should divide n . Q.E.D.

We illustrate the theorem by taking some (n, k) cyclic codes whose $k|n$:

Examples

In the following examples δ and α are taken as primitive elements of $\text{GF}(2^k)$ in NB and in SB respectively, and σ and β are taken as primitive elements of $\text{GF}(2^n)$ in NB and in SB respectively.

Example 5.8.1: $n = 4, k = 2$.

We choose the minimal polynomial for generating $\text{GF}(2^2)$ and $\text{GF}(2^4)$ respectively

as $x^2 + x + 1$ and $x^4 + x + 1$.

The generator polynomial for the code $g(x) = x^2 + 1$, $y = 0101$, $y^2 = 1010$, $y^{2^2} = 0101 = y$. Hence if we choose the basis (generator matrix \underline{G}) as the canonic basis of the form

$\begin{bmatrix} y^2 \\ y \end{bmatrix} = \begin{bmatrix} 1010 \\ 0101 \end{bmatrix}$, then the coefficients can be obtained as:

$$a_{-2^i} = \text{tr}(b_0^{2^i} y), i = 0, 1,$$

where $b_0 = \sum_{j=0}^2 m_{j0} \delta^{-j}$, δ is a primitive element of $GF(2^2)$ in NB. Let α be a primitive element of $GF(2^2)$ in SB. Let σ and β be primitive elements of $GF(2^4)$ in NB and in SB respectively. We choose a NB for $GF(2^2)$ as $\{\alpha, \alpha^2\}$, and for $GF(2^4)$ as $\{\beta^3, \beta^6, \beta^{12}, \beta^9\}$.

$b_0 = \sum_{j=0}^2 m_{j0} \delta^{-j}$ can be calculated using the NB table for $GF(2^2)$ (Table C.1 given in Appendix C), as

$= 1.\delta^{-0} + 1.\delta^{-1} + 0.\delta^{-2} = \delta^0 + \delta^2 = \delta = \sigma^5$ when expressed as a power of a primitive element σ in NB in $GF(2^4)$.

The coefficients can be obtained from

$$a_{-2^i} = \text{tr}(b_0^{2^i} y), i = 0, 1.$$

$y = 0101$, from the NB table for $GF(2^4)$ (Table C.3), is σ^{10} .

Thus the coefficients are

$$a_2 = \text{tr}(b_0 y) = \text{tr}(\sigma^5 \cdot \sigma^{10}) = \text{tr}(\sigma^0) = \sigma^0 + \sigma^0 = 0, \text{ and}$$

$$a_1 = \text{tr}(b_0^2 y) = \text{tr}(\sigma^{10} \cdot \sigma^{10}) = \text{tr}(\sigma^5) = \sigma^5 + \sigma^{10} = \sigma^0 = 1.$$

Thus the LP representing this code in NB corresponding to the canonic basis, is a p-polynomial, say, $f_n(x) = 0.x + 1.x^2 = x^2$.

Note: Even if we take $y = 1010 \neq g(x)$, since $y^2 = 0101$, and $y^{2^2} = 1010 = y$, and since $\{y, y^2\}$ are linearly independent, we again get a p-polynomial in NB, corresponding to the

basis (generator matrix \underline{G}) = $\begin{bmatrix} 0101 \\ 1010 \end{bmatrix}$, whose LP coefficients can be obtained as:

$$a_2 = \text{tr}(b_0 y) = \text{tr}(\sigma^5 \sigma^5) = \text{tr}(\sigma^{10}) = \sigma^{10} + \sigma^5 = \sigma^0 = 1, \text{ and}$$

$$a_1 = \text{tr}(b_0^2 y) = \text{tr}(\sigma^{10} \sigma^5) = \text{tr}(\sigma^0) = \sigma^0 + \sigma^0 = 0.$$

Thus the LP representing this code in NB corresponding to the chosen basis, is a p -polynomial, say, $f_n(x) = 1.x + 0.x^2 = x$.

In total, there are $(2^2-1)(2^2-2) = 6$ ways of choosing a basis for this code. Accordingly there exist 6 different LP representations for the same, two of which were listed above. The remaining 4 NB LP representations can be shown to have coefficients from the subfield $\text{GF}(2^2)$, as per the second part of the theorem, as follows:

$$(1) \quad \underline{G} = \begin{bmatrix} 0101 \\ 1111 \end{bmatrix}, 1111 = \sigma^0, 0101 = \sigma^{10}.$$

$$\begin{bmatrix} a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sigma^5 & \sigma^{10} \\ \sigma^{10} & \sigma^5 \end{bmatrix} \begin{bmatrix} \sigma^0 \\ \sigma^{10} \end{bmatrix} = \begin{bmatrix} 0 \\ \sigma^5 \end{bmatrix}$$

$$\text{Thus } f_n(x) = \sigma^5 x^2.$$

Similarly the other three bases and their NB LP representations are listed below:

$$(2) \quad \underline{G} = \begin{bmatrix} 1010 \\ 1111 \end{bmatrix} \quad f_n(x) = \sigma^{10} x.$$

$$(3) \quad \underline{G} = \begin{bmatrix} 1111 \\ 0101 \end{bmatrix} \quad f_n(x) = \sigma^5 x.$$

$$(4) \quad \underline{G} = \begin{bmatrix} 1111 \\ 1010 \end{bmatrix} \quad f_n(x) = \sigma^{10} x^2.$$

We see in the above 4 cases that the coefficients $\{\sigma^5, \sigma^{10}\} \in \text{GF}(2^4)$ are also elements of the subfield $\text{GF}(2^2)$.

Example 5.8.2: $n = 6, k = 3$.

There are $(2^3 - 2^0)(2^3 - 2^1)(2^3 - 2^2) = 7 \times 6 \times 4 = 168$ different ways of choosing a basis for this cyclic code. However we list only the canonic basis.

Minimal polynomial for generating $GF(2^3)$: $x^3 + x^2 + 1$.

Minimal polynomial for generating $GF(2^6)$: $x^6 + x + 1$.

NB chosen for $GF(2^3)$: $\{\alpha, \alpha^2, \alpha^4\}$.

NB chosen for $GF(2^6)$: $\{\beta^5, \beta^{10}, \beta^{20}, \beta^{40}, \beta^{17}, \beta^{34}\}$.

Generator polynomial of the code $g(x) = x^3 + 1$.

$$y = 001001 = \sigma^{18}, y^2 = 010010 = \sigma^{36}, y^{2^2} = 100100 = \sigma^9, y^{2^3} = 001001 = y.$$

$$\underline{G} = \begin{bmatrix} 100100 \\ 010010 \\ 001001 \end{bmatrix}.$$

$b_0 = \sum_{i=0}^2 m_{j0} \delta^i$ may be calculated using the NB table for $GF(2^3)$ (Table C.2), as

$$b_0 = 1 \cdot \delta^0 + 1 \cdot \delta^1 + 0 \cdot \delta^2 + 1 \cdot \delta^3 + 0 \cdot \delta^4 + 0 \cdot \delta^5 + 1 \cdot \delta^6 = \delta^0 + \delta^6 + \delta^4 + \delta = \delta = \sigma^9.$$

Therefore the coefficients are given by

$$a_6 = \text{tr}(b_0 y) = \text{tr}(\sigma^9 \cdot \sigma^{18}) = \text{tr}(\sigma^{27}) = \sigma^{27} + \sigma^{54} + \sigma^{45} = 0.$$

$$a_5 = \text{tr}(b_0^2 y) = \text{tr}(\sigma^{18} \cdot \sigma^{18}) = \text{tr}(\sigma^{36}) = \sigma^{36} + \sigma^9 + \sigma^{18} = 1.$$

$$a_3 = \text{tr}(b_0^4 y) = \text{tr}(\sigma^{36} \cdot \sigma^{18}) = \text{tr}(\sigma^{54}) = \sigma^{54} + \sigma^{45} + \sigma^{27} = 0.$$

$$\text{Thus } f_n(x) = x^2.$$

Example 5.8.3: $n = 12, k = 4$.

Minimal polynomial for generating $GF(2^4)$: $x^4 + x + 1$.

Minimal polynomial for generating $GF(2^{12})$: $x^{12} + x^6 + x^4 + x + 1$.

NB chosen for $GF(2^4)$: $\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\}$.

NB chosen for $GF(2^{12})$: $\{\beta^{17}, \beta^{34}, \beta^{68}, \beta^{136}, \beta^{272}, \beta^{544}, \beta^{1088}, \beta^{2176}, \beta^{4352}, \beta^{8704}, \beta^{17408}, \beta^{34816}\}$.

Generator polynomial of the code $g(x) = x^8 + x^4 + 1$.

$$y = \sigma^{3549}, y^2 = \sigma^{3003}, y^{2^2} = \sigma^{1911}, y^{2^3} = \sigma^{3822}, y^{2^4} = y.$$

$$\underline{G} = \begin{bmatrix} 100010001000 \\ 010001000100 \\ 001000100010 \\ 000100010001 \end{bmatrix}$$

δ expressed in $GF(2^{12})$ in NB is σ^{273} .

$$b_0 = \delta^{11} = \sigma^{273 \times 11} = \sigma^{3003}.$$

The coefficients are given by

$$a_{14} = \text{tr}(b_0 y) = \text{tr}(\sigma^{3003} \cdot \sigma^{3549}) = \sigma^{2457} + \sigma^{819} + \sigma^{1638} + \sigma^{3276} = 1.$$

$$a_{13} = \text{tr}(b_0^2 y) = \text{tr}(\sigma^{1911} \cdot \sigma^{3549}) = \sigma^{1365} + \sigma^{2730} + \sigma^{1365} + \sigma^{2730} = 0.$$

$$a_{11} = \text{tr}(b_0^4 y) = \text{tr}(\sigma^{3822} \cdot \sigma^{3549}) = \sigma^{3276} + \sigma^{2457} + \sigma^{819} + \sigma^{1638} = 1.$$

$$a_7 = \text{tr}(b_0^8 y) = \text{tr}(\sigma^{3549} \cdot \sigma^{3549}) = \sigma^{3003} + \sigma^{1911} + \sigma^{3822} + \sigma^{3549} = 1.$$

$$\text{Thus } f_n(x) = x^8 + x^4 + x.$$

Example 5.8.4: $n = 10, k = 5$.

Minimal polynomial for generating $GF(2^5)$: $x^5 + x^4 + x^3 + x^2 + 1$.

Minimal polynomial for generating $GF(2^{10})$: $x^{10} + x^3 + 1$.

NB chosen for $GF(2^5)$: $\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$.

NB chosen for $GF(2^{10})$: $\{\beta^7, \beta^{14}, \beta^{28}, \beta^{56}, \beta^{112}, \beta^{224}, \beta^{448}, \beta^{896}, \beta^{769}, \beta^{515}\}$.

Generator polynomial of the code $g(x) = x^5 + 1$.

$$y = \sigma^{561}, y^2 = \sigma^{99}, y^{2^2} = \sigma^{198}, y^{2^3} = \sigma^{396}, y^{2^4} = \sigma^{792}, y^{2^5} = y.$$

$$\underline{G} = \begin{bmatrix} 1000010000 \\ 0100001000 \\ 0010000100 \\ 0001000010 \\ 0000100001 \end{bmatrix}$$

δ expressed in $GF(2^{10})$ in NB is σ^{33} .

$$b_0 = \sum_{i=0}^{30} m_{j0} \delta^{-j} \text{ may be calculated using the NB table for } GF(2^5) \text{ (Table C.4), as}$$

$$\begin{aligned}
b_0 &= \delta^0 + \delta^1 + \delta^5 + \delta^7 + \delta^9 + \delta^{10} + \delta^{12} + \delta^{17} + \delta^{18} + \delta^{21} + \delta^{24} + \delta^{25} + \\
&\quad \delta^{26} + \delta^{27} + \delta^{28} + \delta^{30} = \delta^0 + \delta^{30} + \delta^{26} + \delta^{24} + \delta^{22} + \delta^{21} + \delta^{19} + \delta^{14} + \delta^{13} + \\
&\quad \delta^{10} + \delta^7 + \delta^6 + \delta^5 + \delta^4 + \delta^3 + \delta = \delta^7 = \sigma^{7 \times 33} = \sigma^{231}.
\end{aligned}$$

The coefficients are given by

$$a_{30} = \text{tr}(b_0 y) = \text{tr}(\sigma^{231} \cdot \sigma^{561}) = \sigma^{792} + \sigma^{561} + \sigma^{99} + \sigma^{198} + \sigma^{396} = 1.$$

$$a_{29} = \text{tr}(b_0^2 y) = \text{tr}(\sigma^{462} \cdot \sigma^{561}) = \sigma^0 + \sigma^0 + \sigma^0 + \sigma^0 + \sigma^0 = 1.$$

$$a_{27} = \text{tr}(b_0^4 y) = \text{tr}(\sigma^{924} \cdot \sigma^{561}) = \sigma^{462} + \sigma^{924} + \sigma^{825} + \sigma^{627} + \sigma^{231} = 1.$$

$$a_{23} = \text{tr}(b_0^8 y) = \text{tr}(\sigma^{825} \cdot \sigma^{561}) = \sigma^{363} + \sigma^{726} + \sigma^{429} + \sigma^{858} + \sigma^{693} = 0.$$

$$a_{15} = \text{tr}(b_0^{16} y) = \text{tr}(\sigma^{627} \cdot \sigma^{561}) = \sigma^{165} + \sigma^{330} + \sigma^{660} + \sigma^{297} + \sigma^{594} = 0.$$

$$\text{Thus } f_n(x) = x^4 + x^2 + x.$$

5.9 Decoding of Linear Codes Using GSFs

In this section, we consider applications of GSFs in the decoding of linear (n, k) codes. We interpret a standard array as a two-dimensional (2-D) GSF represented by a two-variable GP. We show how standard array decoders can be implemented using one-dimensional (1-D) and 2-D GSFs.

We also represent syndrome tables using 1-D GSFs where we prove that the GSF representation of a syndrome table is a LFF.

5.9.1 The Standard Array Principle

The standard array is a decoding scheme which partitions all the 2^n n -tuples into 2^k disjoint subsets such that each subset contains only one code vector. The partitioning is done by forming cosets of the linear code. These cosets are disjoint since a linear (n, k) code is a subgroup of the additive group of n -tuples.

There are 2^{n-k} cosets or rows and 2^k columns in a standard array. Every n -tuple (an element of $\text{GF}(2^n)$) appears only once in the array. The leftmost n -tuple in each row is called a *coset leader*. The first row is the code itself, with the all zero vector being its coset

leader. If the received vector, say r (which may be any of the 2^n n -tuples, due to the presence of noise in the channel), is found in one of the 2^k columns, the decoder identifies the transmitted code vector to be that vector which is at the top of the column in which ' r ' is found. Correct decoding results if ' r ' is in the column which corresponds to the actual transmitted code vector.

5.9.2 Representation of Standard Array Using Two-Variable GPs

Standard arrays can be considered as matrices each having 2^{n-k} rows and 2^k columns with elements from $GF(2^n)$, and hence can be compactly represented by two-variable GPs. The coefficients of the GPs are found by first computing the single variable GP coefficients (or Galois Transform) of the rows of the matrix, and replacing the rows with the resulting coefficients, followed by computing the single variable GP coefficients (or Galois Transform) of the resulting columns, to get the final coefficients. We state that the resulting two-variable GP coefficients are nonzero only in the first row and in the first column. This may be argued as follows:

The first row of the standard array contains the code vectors of the corresponding linear code, and the succeeding rows are obtained by adding coset leaders to the code vectors in the first row. Now the single variable GP representing the first row is that of the code itself and hence is a LGP. Let $f_0(x)$ represent this polynomial. Then the succeeding rows will be represented by affine polynomials of the form

$$f_i(x) = a_i + f_0(x), i = 1, 2, \dots, 2^{n-k}-1, \quad (5.9.1)$$

where a_i is the coset leader of the i^{th} row represented as an element of $GF(2^n)$ (For the first row, $a_0 = 0$). Thus when the row transform is computed, we get a LGP, $f_0(x)$, in the first row followed by affine polynomials which differ only in the constant term a_i of $f_0(x)$, in the succeeding rows. Therefore after the row transform computation is over, the matrix will

contain columns each of whose entries are identical except in the first column. Now when the column transform is computed, the first row remains unaffected, since the first row elements become the constant terms in the column transform. Further, only the first column will have a nonzero transform (the coset leader transform) since its elements are different. The remaining columns will be left with zero entries, since DFT of a constant is zero. Thus the two-variable GP representation of a standard array consists of coefficients corresponding to the first row and the first column only.

The coefficients of the two-variable GP representing a standard array belong to $GF(2^L)$ where L is the L.C.M. of n, k and $n-k$. This because while taking the row transform, the mapping is from $GF(2^k)$ to $GF(2^n)$ with coefficients from $GF(2^{L_1})$ where L_1 is the L.C.M. of n and k . Next when the column transform is taken, the mapping is from $GF(2^{n-k})$ to $GF(2^{L_1})$ with the resulting final coefficients lying in $GF(2^L)$ where L is the L.C.M. of $n-k$ and L_1 , which is equal to the L.C.M. of n, k and $n-k$. However, we need not work in that large field, because the first row transform is the LGP representation of the code whose coefficients belong to $GF(2^{L_1})$, L_1 being the L.C.M. of n and k , and the first column transform is the coset leader transform (which is, in general, not a LGP) whose coefficients belong to $GF(2^{L_2})$, L_2 being the L.C.M. of $n-k$ and n . Hence these transforms may be separately determined.

Example 5.9.1: Let us represent the standard array of a linear $(4,2)$ code using two-variable GPs. Let the code vectors be 0000 0101 1010 1111. The standard array for this code is given in Table 5.6a. Let $x^4 + x + 1$ be chosen as the primitive polynomial for generating $GF(2^4)$ and let γ be a primitive element of this field.

Taking the row transform of the above standard array gives transform coefficients belonging to $GF(2^4)$, which are listed in polar form (only the exponents listed) in Table 5.6b.

Table 5.6: Representation of the Standard Array of a Linear (4,2) Code
Using Two-Variable GPs

(a) Standard Array for the Code

0000	0101	1010	1111
0001	0100	1011	1110
0010	0111	1000	1101
0011	0110	1001	1100

(b) Row Transform of the Standard Array in (a)

$-\infty$	$-\infty$	10	1
0	$-\infty$	10	1
1	$-\infty$	10	1
4	$-\infty$	10	1

(c) Column Transform of the Matrix in (b)

$i \downarrow$ $j \rightarrow$	$-\infty$	0	1	2
$-\infty$	$-\infty$	$-\infty$	10	1
0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
1	2	$-\infty$	$-\infty$	$-\infty$
2	8	$-\infty$	$-\infty$	$-\infty$

It may be noted that in the matrix given in Table 5.6b, the rows have identical entries except for the first entry. Now taking the column transform of this matrix, gives the coefficients of the two-variable GP which finally represents the standard array and is shown in Table 5.6c.

The two-variable GP representing the standard array is of the form

$$f(x,y) = \sum_i \sum_j a_{ij} x^{3-i} y^{3-j}$$

where $i, j = -\infty, 0, 1, 2$; $x^{-\infty}, y^{-\infty} = 1$ and the coefficients belong to $GF(2^4)$.

We see that only the first row and first column are nonzero, the first row is the GP representing the code, which is a LGP, and the first column is the GP representing the coset leaders (not a LGP, in general).

Thus $f(x, y)$ in this case, is

$$f(x, y) = \gamma^{10} y^2 + \gamma y + \gamma^2 x^2 + \gamma^8 x.$$

5.9.3 Standard Array Decoding Using GSFs

There are various methods by which a standard array decoder may be implemented for a linear code using GSFs. These methods are described in this subsection with suitable examples.

Decoders can be constructed for given linear (n,k) codes using 1-D as well as 2-D GSFs. In both the cases, we form a mapping table which maps the received n -tuple into an n -tuple code vector or a k -tuple message, according to the standard array for that code, and obtain the GP coefficients representing that mapping. Once the coefficients are obtained, the decoder can be built as a polynomial computer, which computes the transmitted code vector or message by polynomial evaluation.

(i) Using 1-D GSFs

Depending on whether the transmitted code vector (an n -tuple) or the

corresponding message vector (a k -tuple) is required at the receiving end, there can be two ways of implementing a standard array decoder using 1-D GSFs.

(a) Decoding into an n -tuple Code Vector

Suppose we require that the received vector be decoded into the transmitted n -tuple code vector. First we form the standard array. Then using the standard array, we form a mapping table with all the received n -tuples as the domain values and the corresponding code vectors as the range. Now we find the coefficients of the GP representing this mapping. Since this mapping is from $GF(2^n)$ to $GF(2^n)$, the coefficients of the corresponding GP will also belong to $GF(2^n)$. The conjugacy relations are therefore trivial, and the number of terms in the GP is, in general, 2^n . The decoding problem reduces to evaluation of this polynomial at the received vector (an element of $GF(2^n)$). We illustrate this procedure in the following example:

Example 5.9.2: Let us form the single variable decoding polynomial for the $(4,2)$ linear code considered in Example 5.9.1, which decodes the received vector into the transmitted vector according to its standard array.

Let $x^4 + x + 1$ be the primitive polynomial for generating $GF(2^4)$ and let α be a primitive element of this field. Then using the standard array of this code given in Table 5.6a, we form a mapping table with the received n -tuples taken in the order $\alpha^{-\infty}$, α^0 , α , ..., α^{14} , as the domain values, and the corresponding code vectors (which are the elements in the top row of the standard array) as the range. This is shown in Table 5.7. It may be noted that, in the polar notation of Table 5.7, only the exponents of α are listed. The GP representing this mapping may be obtained as

$$f(x) = \alpha^{13} x^2 + \alpha^8 x^4 + \alpha^3 x^8.$$

Table 5.7: Decoding Table for the Linear (4,2) Code considered in Example 5.9.1
with the Received n-tuples as Domain and the Transmitted n-tuples as Range

Received n-tuple		Transmitted n-tuple	
Polar	Cartesian	Polar	Cartesian
$-\infty$	0 0 0 0	$-\infty$	0 0 0 0
0	0 0 0 1	$-\infty$	0 0 0 0
1	0 0 1 0	$-\infty$	0 0 0 0
2	0 1 0 0	8	0 1 0 1
3	1 0 0 0	9	1 0 1 0
4	0 0 1 1	$-\infty$	0 0 0 0
5	0 1 1 0	8	0 1 0 1
6	1 1 0 0	12	1 1 1 1
7	1 0 1 1	9	1 0 1 0
8	0 1 0 1	8	0 1 0 1
9	1 0 1 0	9	1 0 1 0
10	0 1 1 1	8	0 1 0 1
11	1 1 1 0	12	1 1 1 1
12	1 1 1 1	12	1 1 1 1
13	1 1 0 1	12	1 1 1 1
14	1 0 0 1	9	1 0 1 0

Now let us assume that the received vector is, say 1 1 0 0. From the table, this is equal to α^6 . Evaluating $f(x)$ at α^6 , we get

$$f(\alpha^6) = \alpha^{13} \cdot \alpha^{12} + \alpha^8 \cdot \alpha^9 + \alpha^3 \cdot \alpha^3 = \alpha^{10} + \alpha^2 + \alpha^6 = \alpha^{12} = 1 1 1 1.$$

Thus 1 1 0 0 is decoded into 1 1 1 1. This may be verified from the decoding table given in Table 5.7.

(b) Decoding into a k-tuple Message Vector

A second possibility of decoding using 1-D GSFs would be to decode the received vector directly into the corresponding k-tuple message vector instead of the n-tuple code vector. Thus we first form a mapping table with the domain as the received n-tuple vectors (from $GF(2^n)$) and the range as the message k-tuple vectors corresponding to the n-tuple code vectors, and then find the coefficients of the GP representing this mapping. Here the decoding polynomial thus obtained will have coefficients from $GF(2^L)$, (where L is the L.C.M. of n and k), the same field as that of the encoding polynomial. The decoding polynomial will not be, in general, linearized, unlike the encoding polynomial. However conjugacy relations always exist, since now the range values belong to $GF(2^k)$, a proper subfield of $GF(2^L)$, where the coefficients lie. Thus the decoding polynomial is a Frobenius polynomial (FP) expressed as the sum of various Frobenius terms. The decoding problem, in this case, reduces to the computation of Frobenius sums which can be efficiently carried out in NB representation. Further, since each Frobenius term in the FP may be evaluated independently, this makes possible a parallel implementation scheme for the decoder.

We illustrate this method of decoding in the following examples:

Example 5.9.3: Let us take a (4,2) linear code with the code vectors being 0000 1111 0111 1000. The corresponding message k-tuples are respectively 00 01 10 11. Let the primitive polynomial for generating $GF(2^4)$ be $x^4 + x + 1$. Then the subfield $GF(2^2)$ is generated by the minimal polynomial of α^5 which is equal to $x^2 + x + 1$, where α is a primitive element of $GF(2^4)$. Let β be a primitive element of $GF(2^2)$. Using the standard array in Table 5.8a, we form a mapping table with the received n-tuples taken in the order $\alpha^{-\infty}, \alpha^0, \alpha, \dots, \alpha^{14}$, as the domain values, and the corresponding message k-tuples (instead of the code vectors) belonging to $GF(2^2)$ as the range. This is given in Table 5.8b.

Table 5.8 Decoding of a Linear (4,2) Code into k-tuple Message Vectors
using 1-D GSFs

(a) Standard Array for the Code

0 0 0 0	1 1 1 1	0 1 1 1	1 0 0 0
0 0 0 1	1 1 1 0	0 1 1 0	1 0 0 1
0 0 1 0	1 1 0 1	0 1 0 1	1 0 1 0
0 1 0 0	1 0 1 1	0 0 1 1	1 1 0 0

(b) Decoding Table for the Code with the Received n-tuples as Domain
and transmitted k-tuples as Range

Received n-tuple		Transmitted n-tuple	
Polar	Cartesian	Polar	Cartesian
$-\infty$	0 0 0 0	$-\infty$	0 0
0	0 0 0 1	$-\infty$	0 0
1	0 0 1 0	$-\infty$	0 0
2	0 1 0 0	$-\infty$	0 0
3	1 0 0 0	2	1 1
4	0 0 1 1	1	1 0
5	0 1 1 0	1	1 0
6	1 1 0 0	2	1 1
7	1 0 1 1	0	0 1
8	0 1 0 1	1	1 0
9	1 0 1 0	2	1 1
10	0 1 1 1	1	1 0
11	1 1 1 0	0	0 1
12	1 1 1 1	0	0 1
13	1 1 0 1	0	0 1
14	1 0 0 1	2	1 1

It may be noted that, in the polar notation of Table 5.8b, only the exponents of α and β are respectively listed

The GP representing this mapping may be obtained as

$$f(x) = \alpha^8 x + \alpha^3 x^2 + x^3 + \alpha^2 x^4 + x^5 + \alpha^{10} x^6 + \alpha^{12} x^8 + \alpha^{10} x^9 + \alpha^{10} x^{10} + x^{12}.$$

The above expression may be expressed as a sum of Frobenius terms as

$$f(x) = \text{frs}(\alpha^8 x) + \text{frs}(\alpha^3 x^2) + \text{frs}(x^3) + \text{frs}(x^5) + \text{frs}(\alpha^{10} x^6) + \text{frs}(\alpha^{10} x^{10}),$$

where $\text{frs}(\Theta) = \Theta + \Theta^4$, except in the case of $\text{frs}(x^5)$ and $\text{frs}(\alpha^{10} x^{10})$, which are respectively x^5 and $\alpha^{10} x^{10}$.

Now let the received n -tuple be $1\ 0\ 1\ 1 = \alpha^7$. Then

$$f(\alpha^7) = \text{frs}(\alpha^8 \alpha^7) + \text{frs}(\alpha^3 \alpha^{14}) + \text{frs}(\alpha^6) + \text{frs}(\alpha^5) + \text{frs}(\alpha^{10} \alpha^{12}) + \text{frs}(\alpha^{10} \alpha^{10}) = (\alpha^0 + \alpha^0) + (\alpha^2 + \alpha^8) + (\alpha^6 + \alpha^9) + (\alpha^5) + (\alpha^7 + \alpha^{13}) + (\alpha^5) = \alpha^0 = \beta^0 = 0\ 1. \text{ Thus the transmitted message is } 0\ 1, \text{ which may be verified to be true from Table 5.8b.}$$

Example 5.9.4: Let us take the example of a linear $(5,2)$ code with the code vectors 00000 01101 10111 11010 . The corresponding message k -tuples are respectively 00 01 10 11 . Since the L.C.M. of 5 and 2 is 10, the coefficients of the decoding polynomial belong to $\text{GF}(2^{10})$. Let the primitive polynomial for generating $\text{GF}(2^{10})$ be $x^{10} + x^7 + 1$. Let the primitive elements of $\text{GF}(2^{10})$, $\text{GF}(2^5)$ and $\text{GF}(2^2)$ be chosen as γ , α and β respectively. Then the subfields $\text{GF}(2^5)$ and $\text{GF}(2^2)$ are respectively generated by the minimal polynomials of γ^{33} and γ^{341} , which are $x^5 + x^3 + x^2 + x + 1$ and $x^2 + x + 1$ respectively.

The standard array for this code may be formed as in Table 5.9.

We then form a mapping table with domain values from $\text{GF}(2^5)$, which are all the received 5-tuples, in the order $\alpha^{-\infty}$, α^0 , α , ..., α^{30} , and the range values from $\text{GF}(2^2)$ which are the message 2-tuples corresponding to the code vectors.

Table 5.9: Standard Array for the Linear (5,2) Code of Example 5.9.4

0 0 0 0 0	0 1 1 0 1	1 0 1 1 1	1 1 0 1 0
0 0 0 0 1	0 1 1 0 0	1 0 1 1 0	1 1 0 1 1
0 0 0 1 0	0 1 1 1 1	1 0 1 0 1	1 1 0 0 0
0 0 1 0 0	0 1 0 0 1	1 0 0 1 1	1 1 1 1 0
0 1 0 0 0	0 0 1 0 1	1 1 1 1 1	1 0 0 1 0
1 0 0 0 0	1 1 1 0 1	0 0 1 1 1	0 1 0 1 0
0 0 0 1 1	0 1 1 1 0	1 0 1 0 0	1 1 0 0 1
1 0 0 0 1	1 1 1 0 0	0 0 1 1 0	0 1 0 1 1

The GP representing this mapping, expressed as a sum of Frobenius terms, is

$$f(x) = \text{frs}(\gamma^{483} x) + \text{frs}(\gamma^{799} x^3) + \text{frs}(\gamma^{90} x^5) + \text{frs}(\gamma^{297} x^7) + \text{frs}(\gamma^{792} x^{11}),$$

$$\text{where } \text{frs}(\Theta) = \Theta + \Theta^4 + \Theta^{16} + \Theta^{64} + \Theta^{256}.$$

Let the received n-tuple be 0 0 1 1 1 = $\alpha^{27} = \gamma^{891}$. Then

$$\begin{aligned} f(\alpha^{27}) &= f(\gamma^{891}) = \text{frs}(\gamma^{483} \cdot \gamma^{891}) + \text{frs}(\gamma^{799} \cdot \gamma^{627}) + \text{frs}(\gamma^{90} \cdot \gamma^{363}) + \text{frs}(\gamma^{297} \cdot \gamma^{99}) \\ &\quad + \text{frs}(\gamma^{792} \cdot \gamma^{594}) = \text{frs}(\gamma^{351}) + \text{frs}(\gamma^{403}) + \text{frs}(\gamma^{453}) + \text{frs}(\gamma^{396}) + \text{frs}(\gamma^{363}) \\ &= (\gamma^{351} + \gamma^{381} + \gamma^{501} + \gamma^{981} + \gamma^{855}) + (\gamma^{403} + \gamma^{589} + \gamma^{310} + \gamma^{217} + \gamma^{868}) \\ &\quad + (\gamma^{453} + \gamma^{789} + \gamma^{87} + \gamma^{348} + \gamma^{369}) + (\gamma^{396} + \gamma^{561} + \gamma^{198} + \gamma^{792} + \gamma^{99}) \\ &\quad + (\gamma^{363} + \gamma^{429} + \gamma^{693} + \gamma^{726} + \gamma^{858}) = \gamma^{341} = \beta = 10. \end{aligned}$$

Thus the received n-tuple 0 0 1 1 1, is decoded into the message k-tuple 1 0.

(ii) Using 2-D GSFs

Here the idea is to split each received n-tuple into an n_1 -tuple and n_2 -tuple such that $n = n_1 + n_2$. A decoding table is formed using the standard array, with the domain values from $\text{GF}(2^{n_1})$ and $\text{GF}(2^{n_2})$ and the range values from either $\text{GF}(2^n)$ (the code vectors) or $\text{GF}(2^k)$ (the message k-tuples). A two-variable GP, say $f(x, y)$, representing this mapping whose coefficients belong to $\text{GF}(2^L)$, where L is the L.C.M. of n_1 , n_2 and n or

k (as the case may be), can then be found. If the requirement is to decode into message k -tuples, a suitable choice for n_1 and n_2 are k and $n-k$ respectively, in which case, the extension order L of $GF(2^L)$ is confined to the L.C.M. of k and $n-k$.

Given a received n -tuple vector, decoding is done by splitting it into an n_1 -tuple and an n_2 -tuple, finding their polar representation x and y in the respective fields (ie., $GF(2^{n_1})$ and $GF(2^{n_2})$ respectively) and computing the polynomial value at this x and y .

We illustrate the procedure of decoding using 2-D GSFs into n -tuples and k -tuples respectively in the following examples:

(a) Decoding into an n -tuple Code Vector

Example 5.9.5. Let us assume that a two-variable decoding polynomial which decodes the received vector into the corresponding n -tuple code vector, is required. We take the linear $(4,2)$ code considered in Example 5.9.3 with the code vectors being 0000 1111 0111 1000.

We split each 4-tuple into two 2-tuples. The coefficients of the two-variable decoding polynomial would belong to $GF(2^4)$. Let $x^4 + x + 1$ be a primitive polynomial for generating $GF(2^4)$. Then the subfield $GF(2^2)$ is generated by the minimal polynomial of $\alpha = \gamma^{2^1}$, which is $x^2 + x + 1$, where α and γ are primitive elements of $GF(2^2)$ and $GF(2^4)$ respectively. Now with elements from $GF(2^2)$ as domain values, and the n -tuple code vectors as the range, we form a decoding table as in Table 5.10a. It may be noted that, in the table, x and y values both belong to $GF(2^2)$ and they are in the order of the power of a primitive element.

The two-variable GP representing this decoding table is of the form

$$f(x,y) = \sum_i \sum_j a_{ij} x^{3-i} y^{3-j},$$

where $i, j = \infty, 0, 1, 2$; $x^\infty, y^\infty = 1$ and the coefficients belong to $GF(2^4)$.

The coefficients a_{ij} are listed as a power of γ in Table 5.10b (Only the exponents of γ are listed).

Conjugacy relations do not exist among the coefficients. The decoding polynomial may be expressed as

$$f(x, y) = \gamma^{10} y^3 + \gamma^5 y^2 + y + \gamma^3 x^2 + \gamma^{10} x^2 y^2 + \gamma^5 x^2 y + \gamma^3 x + x y^2 + \gamma^{10} x y.$$

Now let the received vector be, say 1 0 1 1. We split it into two 2-tuples, 1 0 = $\alpha = \gamma^5$, and 1 1 = $\alpha^2 = \gamma^{10}$. Substituting $x = \gamma^5$ and $y = \gamma^{10}$ in $f(x, y)$, we get

$$f(x, y) = \gamma^{10} + \gamma^{10} + \gamma^{10} + \gamma^{13} + \gamma^{10} + \gamma^{10} + \gamma^8 + \gamma^{10} + \gamma^{10} = \gamma^{12} = 1 1 1 1.$$

Thus 1 0 1 1 is decoded into the code vector 1 1 1 1.

Table 5.10: Decoding of a Linear (4,2) Code into n-tuple Code Vectors

Using 2-D GSFs

(a) Decoding Table with the Received n-tuples (split into k-tuples and n-k tuples) as Domain and the Transmitted n-tuples as Range

$\begin{matrix} x \rightarrow \\ y \rightarrow \\ \downarrow \end{matrix}$	0 0	0 1	1 0	1 1
0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 1 1 1
0 1	0 0 0 0	0 1 1 1	0 1 1 1	0 1 1 1
1 0	1 0 0 0	1 0 0 0	1 0 0 0	1 1 1 1
1 1	1 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1

(b) Coefficients of the Two-Variable GP Representing (a)

$\begin{matrix} i \\ \downarrow \\ j \end{matrix}$	$-\infty$	0	1	2
$-\infty$	$-\infty$	10	5	0
0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
1	3	$-\infty$	10	5
2	3	$-\infty$	0	10

(b) Decoding into a k-tuple Message Vector

Example 5.9.6: Let us take the (5,2) linear code which was given in Example 5.9.4, and let us assume that the received vector is to be decoded into a k-tuple message using 2-D GSFs. We first split each 5-tuple into a 2-tuple and a 3-tuple. Then the coefficients of the two-variable GP which performs the decoding process belong to $GF(2^6)$. Let $x^6 + x + 1$ be a primitive polynomial for generating $GF(2^6)$. Then the subfields $GF(2^3)$ and $GF(2^2)$ are generated by the minimal polynomials of $\alpha = \gamma^9$ and $\beta = \gamma^{21}$ respectively, which are $x^3 + x^2 + 1$ and $x^2 + x + 1$, where γ , α and β are primitive elements of $GF(2^6)$, $GF(2^3)$ and $GF(2^2)$ respectively. Now with elements from $GF(2^3)$ and $GF(2^2)$ as the domain values, and the message k-tuples as the range, we form the decoding table in Table 5.11a. As in Example 5.9.5, x values belong to $GF(2^3)$ and y values belong to $GF(2^2)$ in the table, and they are in the order of the power of a primitive element in the respective fields.

The two-variable GP representing this decoding table is of the form

$$f(x, y) = \sum_i \sum_j a_{ij} x^{7-i} y^{3-j},$$

where $i = -\infty, 0, 1, \dots, 6$; $j = -\infty, 0, \dots, 2$; $x^{-\infty}, y^{-\infty} = 1$ and the coefficients belong to $GF(2^6)$.

The coefficients a_{ij} are listed as a power of γ in Table 5.11b (Only the exponents of γ are listed).

We may note that conjugacy relations exist among the coefficients and hence the two-variable decoding polynomial can be expressed as a sum of Frobenius terms. The coefficients satisfy the conjugacy relations as follows:

$$(a_{ij})^4 = a_{4.i \bmod 7, 4.j \bmod 3}.$$

$$i = -\infty, 0, 1, \dots, 6, j = -\infty, 0, 1, 2.$$

Thus the decoding polynomial may be expressed as

$$f(x, y) = x^7 + \text{frs}(\gamma^4 x^6) + \text{frs}(\gamma^{56} x^4) + \text{frs}(x^4 y^3) + \text{frs}(\gamma^{56} x^6 y^2) + \text{frs}(\gamma^{55} x^4 y^2) + \text{frs}(\gamma^7 x^6 y) + \text{frs}(\gamma^5 x^4 y), \text{ where } \text{frs}(\Theta) = \Theta + \Theta^4 + \Theta^2.$$

Table 5.11: Decoding of a Linear (5,2) Code into k-tuple Message Vectors
Using 2-D GSFs

(a) Decoding Table with the Received n-tuples (split into k-tuples and n-k tuples)
as Domain and the Transmitted k-tuples as Range

$\begin{matrix} x & y \\ \downarrow & \end{matrix}$	0 0	0 1	1 0	1 1
0 0 0	0 0	0 0	0 0	0 0
0 0 1	0 0	0 1	1 0	1 0
0 1 0	0 0	0 1	1 1	1 1
1 0 0	0 0	0 0	1 1	1 0
1 0 1	1 0	1 0	1 0	1 0
1 1 1	0 1	0 1	1 1	1 0
0 1 1	0 1	0 1	0 1	0 1
1 1 0	1 1	1 1	1 1	1 1

(b) Coefficients of the Two-Variable GP Representing (a)

$\begin{matrix} i & j \\ \downarrow & \end{matrix}$	$-\infty$	0	1	2
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
0	0	$-\infty$	$-\infty$	$-\infty$
1	4	$-\infty$	56	7
2	1	$-\infty$	14	49
3	56	0	55	5
4	16	$-\infty$	35	28
5	35	0	31	20
6	14	0	61	17

Now let the received vector be, say 1 1 0 1 0. We split it into a 3-tuple $1 1 0 = \alpha^6 = \gamma^{54}$,
and a 2-tuple $1 0 = \beta = \gamma^{21}$. Substituting $x = \gamma^{54}$ and $y = \gamma^{21}$ in $f(x, y)$, we get
$$f(x, y) = \gamma^0 + \text{frs}(\gamma^{13}) + \text{frs}(\gamma^{20}) + \text{frs}(\gamma^{27}) + \text{frs}(\gamma^{44}) + \text{frs}(\gamma^{61}) + \text{frs}(\gamma^{37}) + \text{frs}(\gamma^{53})$$

$= \gamma^0 + \gamma^0 + \gamma^{42} + 0 + \gamma^{42} + \gamma^{42} + \gamma^{21} + \gamma^{21} = \gamma^{42} = \beta^2 = 11$. Thus 11010 is decoded into the message vector 11.

We conclude this chapter with a description of the GP representation of a syndrome table which plays an important role in standard array decoding.

5.9.4 Syndrome Tables and their Representation Using GPs

Syndrome is an important parameter of a standard array, which helps in simplifying the decoding process. If \underline{H} is the parity check matrix of a linear (n, k) code, of size $(n-k, n)$, and if \underline{r} is a received n -tuple, then syndrome \underline{S} is defined as

$$\underline{S} = \underline{r} \underline{H}^T, \quad (5.9.2)$$

where T denotes transpose.

It may be noted that all the 2^k n -tuples of a coset have the same syndrome. Further, syndromes for different cosets are different. The syndrome of any code vector is equal to zero.

In this section, we find the GP representation of a syndrome table. We note that a syndrome table is a many-to-one mapping from $GF(2^n)$ to $GF(2^{n-k})$. Hence it can be represented by a GP with coefficients from $GF(2^{L_2})$, L_2 being the L.C.M. of n and $n-k$. We state that this GP is a LGP. This is because the syndrome \underline{S} , given by (5.9.2), is obtained by a linear combination of the rows of \underline{H}^T . Further, since $n-k$ is smaller than n , $GF(2^{n-k})$ is a proper subfield of $GF(2^{L_2})$, and thus the coefficients satisfy conjugacy constraints resulting in the associated LGP being an LFP.

All the roots of the LFP representing a syndrome table may not lie in $GF(2^n)$, but may belong to an extension of it also. *The roots in $GF(2^n)$, of the LFP representing the syndrome table, give the code vectors of the corresponding linear (n, k) code*, because the value of this LFP at a code vector is equal to zero [In Chapter 6, we will study the properties and applications of special types of GSFs represented by LPs known as *syndrome polynomials*, all of whose roots lie in $GF(2^n)$ and can be used to uniquely

characterize linear (n,k) codes of a given pair of n and k].

Example 5.9.7. Let us take the linear $(5,2)$ code considered in Example 5.9.4. A basis for this code is given by the generator matrix $\underline{G} = \begin{bmatrix} 10111 \\ 01101 \end{bmatrix}$. Then the parity check matrix for this code may be obtained as $\underline{H} = \begin{bmatrix} 11100 \\ 10010 \\ 11001 \end{bmatrix}$. By taking all possible linear combinations of the matrix \underline{H}^T , we get a mapping from $GF(2^5)$ to $GF(2^3)$. If we find the GP representing this mapping, we get a LGP, say $f_s(x)$, which represents the syndrome table. The value of $f_s(x)$ at any received n -tuple, gives the syndrome corresponding to that n -tuple.

Since the L.C.M. of 5 and 3 is 15, the coefficients of $f_s(x)$ belong to $GF(2^{15})$. Let the primitive polynomial for generating $GF(2^{15})$ be $x^{15} + x + 1$. Let the primitive elements of $GF(2^{15})$, $GF(2^5)$ and $GF(2^3)$ be chosen as γ , α and β respectively. Then the subfields $GF(2^5)$ and $GF(2^3)$ are respectively generated by the minimal polynomials of γ^{1057} and γ^{4681} , which are $x^5 + x^3 + x^2 + x + 1$ and $x^3 + x + 1$ respectively.

$f_s(x)$ may be obtained as

$$f_s(x) = \gamma^{15884} x + \gamma^{799} x^2 + \gamma^{18369} x^4 + \gamma^{28771} x^8 + \gamma^{6392} x^{16}.$$

This may be expressed as a single term LFP as

$$f_s(x) = \text{frs}(\gamma^{15884} x),$$

where the Frobenius sum is taken with respect to $GF(2^3)$.

CHAPTER 6

SYNDROME POLYNOMIAL REPRESENTATIONS OF LINEAR BLOCK CODES

In this chapter, we consider a special class of Galois switching functions (GSFs) from $GF(2^n)$ to $GF(2^n)$ which are represented by Galois polynomials (GPs) satisfying the following relations:

$$\begin{aligned} S(x) &= 0 ; & x \in U \\ &= y ; & x \notin U, y \in U^d \end{aligned} \quad (6.1a)$$

where U is a k -dimensional subspace of $GF(2^n)$, and U^d is an $(n-k)$ dimensional subspace of $GF(2^n)$ which satisfies another GP, representing a GSF from $GF(2^n)$ to $GF(2^n)$, given by

$$\begin{aligned} S_d(x) &= 0 ; & x \in U^d \\ &= y ; & x \notin U^d, y \in U \end{aligned} \quad (6.1b)$$

From the theory of linearized polynomials (LPs) (described in Appendix A), it follows that $S(x)$ is a LP and $S_d(x)$ is its dual LP. The GSFs of the kind considered in this chapter provide an alternative representation of linear (n,k) block codes in terms of LPs by considering the subspace structure of linear block codes and forming polynomials with the code vectors as roots. We call these polynomials as *syndrome polynomials* (SPs) for reasons which will become evident in Section 6.2.

We further show that SPs represented in normal basis (NB) have interesting properties which help in the characterization of well known linear block codes such as quasi cyclic codes and cyclic codes and that they lead to new methods for the study of weight distributions of such codes.

In all our discussions, the term 'linear code' or a 'linear (n,k) code' implies a binary linear (n,k) block code, wherever it is used unless otherwise stated.

6.1 Representation of a Linear Code as the Root Space of a Linearized Polynomial

A linear (n,k) code is a k-dimensional subspace U of the vector space of n-tuples. The 2^k code vectors of this code are elements of $GF(2^n)$. We form the polynomial

$S(x) = \prod_{\beta \in U} (x - \beta)$ where β are the 2^k code vectors belonging to $GF(2^n)$. This polynomial is a monic LP of degree 2^k . Thus we have the following theorem relating linear codes with the roots of appropriate LPs:

Theorem 6.1.1: Any linear (n,k) code U can be uniquely represented by a monic LP over $GF(2^n)$ of degree 2^k , of the form

$$S(x) = \prod_{\beta \in U} (x - \beta) = x^{2^k} + a_{k-1} x^{2^{k-1}} + \dots + a_0 x, \quad (6.1.1)$$

where β is a code vector considered as an element of $GF(2^n)$ belonging to U, and $a_0 \neq 0$.

Proof: We directly apply Theorem A.3.2 with the comment that the linear (n,k) code is a k-dimensional subspace of $GF(2^n)$. Further, if $a_0 = 0$, then $S(x)$ has multiple roots meaning that the code vectors are not uniquely assigned to the message vectors, which is not true. Hence $a_0 \neq 0$. Q.E.D.

We also have a converse to the above theorem:

Theorem 6.1.2: Any monic LP, $S(x)$, of degree 2^k of the form (6.1.1), with $a_0 \neq 0$, which divides $x^{2^n} - x$, represents a linear (n,k) code, the code vectors being members of the root space of $S(x)$.

Proof: Let $S(x)$ be a monic LP of degree 2^k with $a_0 \neq 0$, which divides $x^{2^n} - x$. Now, the splitting field for $x^{2^n} - x$, i.e., the field where all the roots of $x^{2^n} - x$ are lying, is $GF(2^n)$. Since $S(x)$ divides $x^{2^n} - x$, all the roots of $S(x)$ lie in $GF(2^n)$. Further, as $S(x)$ is a LP of degree 2^k , its roots form a k -dimensional subspace of n -tuples and since $a_0 \neq 0$, the roots are simple. In other words, the code vectors are members of the root space of $S(x)$.
Q.E.D.

Theorem 6.1.3 The number of LPs of the form (6.1.1) which divide $x^{2^n} - x$ is given by

$$N_{\text{dist}} = N_n / N_k,$$

where

$$N_n = (2^n - 1)(2^n - 2)(2^n - 2^2) \dots (2^n - 2^{k-1}) \text{ and}$$

$$N_k = (2^k - 1)(2^k - 2)(2^k - 2^2) \dots (2^k - 2^{k-1}).$$

Proof: The number of distinct binary linear codes of a given pair of n and k is equal to N_{dist} as given in Chapter 3. Since there is a one-to-one correspondence between LPs of the form (6.1.1) and linear codes of a given pair of n and k , the number of such LPs is equal to the number of distinct linear codes of a given pair of n and k .
Q.E.D.

6.2 Linearized Polynomials for Decoding of Linear Codes

We call the LPs whose roots are the code vectors of a linear (n, k) code, as *syndrome polynomials* (SPs). This is because the structure of these polynomials may be exploited for the decoding of linear codes. We show that SPs can be used for computing syndromes of respective linear codes.

A LP has the property that its range space can have values only from the root space of its dual polynomial. The SP representing a linear (n, k) code, say $S(x)$, is of degree 2^k , and its dual polynomial, say $S_d(x)$, is of degree 2^{n-k} . Thus the dual polynomial has 2^{n-k} roots which appear as the range values of $S(x)$. Each root of $S_d(x)$ appears in the range

In other words, any received vector r_i in the j^{th} coset will have the same range value $S(t_j)$, which acts as a syndrome for that coset.

Q.E.D.

Thus given any received vector r , an element of $GF(2^n)$, the procedure for decoding would be as follows.

Compute the n -tuple syndrome $S(r)$. Identify the coset leader t corresponding to this syndrome. Adding t to r , gives the transmitted code vector.

Example 6.2.1: Let us take the example of a linear (5,2) code with the code vectors 00000, 10100, 11111, 01011. Let $x^5 + x^2 + 1$ be a primitive polynomial for generating $GF(2^5)$ with α as a primitive element. Then the above code vectors expressed as a power of α , is $\alpha^{-\infty}$, α^7 , α^{15} and α^{27} respectively. The SP for this linear code may be computed as

$$S(x) = x^4 + \alpha^9 x^2 + \alpha^{18} x.$$

This polynomial assumes the range values α^5 , α^7 , α^{10} , α^{14} , α^{20} , α^{21} and α^{29} , which are the roots of its dual polynomial $S_d(x)$ given by

$$S_d(x) = x^8 + \alpha^{10} x^4 + \alpha^{17} x^2 + \alpha^{13} x.$$

The standard array for this code may be formed as in Table 6.1.

Table 6.1: Standard Array for a Linear (5,2) Code

0 0 0 0 0	1 0 1 0 0	1 1 1 1 1	0 1 0 1 1
0 0 0 0 1	1 0 1 0 1	1 1 1 1 0	0 1 0 1 0
0 0 0 1 0	1 0 1 1 0	1 1 1 0 1	0 1 0 0 1
0 0 1 0 0	1 0 0 0 0	1 1 0 1 1	0 1 1 1 1
0 1 0 0 0	1 1 1 0 0	1 0 1 1 1	0 0 0 1 1
0 0 1 1 0	1 0 0 1 0	1 1 0 0 1	0 1 1 0 1
0 1 1 0 0	1 1 0 0 0	1 0 0 1 1	0 0 1 1 1
0 0 1 0 1	1 0 0 0 1	1 1 0 1 0	0 1 1 1 0

In Table 6.2, we give the above array with the entries expressed as a power of α . We also list the range value, $S(r)$, for each coset, in the first column, which acts as a syndrome.

Only the exponents of α are listed in either case.

Table 6.2: Standard Array of Table 6.1 including Syndromes (expressed in Polar Form)

$S(r)$				
$-\infty$	$-\infty$	7	15	27
21	0	22	24	6
10	1	28	14	29
14	2	4	16	23
29	3	13	26	18
20	19	30	25	8
7	20	21	17	11
5	5	10	9	12

Now as an example of decoding using $S(x)$, let us assume that the transmitted code vector c is 1 0 1 0 0, and let the second bit be corrupted so that the received vector r is 1 1 1 0 0 = α^{13} . We compute the syndrome by substituting $x = \alpha^{13}$ in $S(x)$. We get the syndrome as $S(\alpha^{13}) = (\alpha^{13})^4 + \alpha^9(\alpha^{13})^2 + \alpha^{18}(\alpha^{13}) = \alpha^{21} + \alpha^4 + 1 = \alpha^{29}$. Now from the table, it is seen that the coset leader t corresponding to this syndrome is $\alpha^3 = 0 1 0 0 0$. Therefore we get the transmitted vector c as $r + t = 1 1 1 0 0 + 0 1 0 0 0 = 1 0 1 0 0$.

6.3 Normal Basis Syndrome Polynomials

SPs represented with respect to an appropriate *normal basis* (NB) have a special significance in coding theory for the following reasons:

First, this helps us in the classification of linear codes on the basis of their weight distributions. Secondly, the class of t -cyclic codes (quasi-cyclic codes which are closed under t cyclic shifts, $t \geq 1$) may be completely characterized by their SP representations in NB. Thirdly, and most importantly, cyclic codes ($t = 1$) have unique SP representations in the form of *p-polynomials* in NB. Further, the problem of finding weight distributions in cyclic codes may now be reduced to the problem of factorization of their NB *p-polynomial*

representations. Finally, self dual cyclic codes and other well known cyclic codes like Bose—Chaudhuri—Hocquenghem (BCH) and Golay codes may be compactly represented using NB p -polynomials and their weight distributions studied.

In the following sections, we bring out the above facts in detail. Hereafter, we assume that the code vectors are recognized as elements in some NB of $GF(2^n)$.

6.4 Normal Basis Syndrome Polynomial Representations of Linear Codes with the Same Weight Distribution

The following theorem gives a result relating codes with the same weight distribution:

Theorem 6.4.1: If the monic LP

$$S_0(x) = x^{2^k} + a_{k-1} x^{2^{k-1}} + \dots + a_0 x \quad (6.4.1)$$

represents the SP of a linear (n,k) code, then the linear (n,k) codes represented by the SPs

$$S_j(x) = x^{2^k} + (a_{k-1})^{2^j} x^{2^{k-1}} + (a_{k-2})^{2^j} x^{2^{k-2}} + \dots + (a_0)^{2^j} x, \quad (6.4.2)$$

where $j = 0, 1, 2, \dots, t-1$, $(a_i)^{2^t} = a_i$, $i = 0, 1, \dots, k-1$, $a_0 \neq 0$, have the same weight distribution, if a_i 's are elements belonging to some NB of $GF(2^n)$.

Proof: Let the roots of $S_0(x)$ be $r_0, r_1, r_2, \dots, r_\rho$ where $\rho = 2^k - 1$ and each r_i is a code vector of a linear (n,k) code C_0 , considered as elements belonging to some NB of $GF(2^n)$.

Let r_0 be the zero code vector.

Now let us consider linear (n,k) codes, C_j , whose code vectors are obtained by cyclic shifting each of the n -tuple code vectors of C_0 represented in some NB of $GF(2^n)$, by j places where $j = 1, 2, \dots, t-1$, and t cyclic shifts results in the code C_0 itself.

Evidently, the set of codes C_j , $j = 0, 1, \dots, t-1$, will have the same weight

distribution. Let us now consider the SPs representing the codes in this set. Cyclic shifting of the code vectors r_i , $i = 0, 1, 2, \dots, 2^k-1$, by j places is equivalent to raising r_i to 2^j th power, in NB. It is known that [27], if r_1, r_2, \dots, r_k form a basis for C_0 , the monic LP representing C_0 is given by

$$S_0(x) = D(x)/D_k, \quad (6.4.3)$$

where D_k is the determinant given by

$$D_k = \begin{vmatrix} r_1 & r_1^2 & r_1^{2^2} & \dots & r_1^{2^{k-1}} \\ r_2 & r_2^2 & r_2^{2^2} & \dots & r_2^{2^{k-1}} \\ \dots & \dots & \dots & \dots & \dots \\ r_k & r_k^2 & r_k^{2^2} & \dots & r_k^{2^{k-1}} \end{vmatrix}$$

and $D(x)$ is the polynomial given by

$$D(x) = \begin{vmatrix} r_1 & r_1^2 & r_1^{2^2} & \dots & r_1^{2^{k-1}} & r_1^{2^k} \\ r_2 & r_2^2 & r_2^{2^2} & \dots & r_2^{2^{k-1}} & r_2^{2^k} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ r_k & r_k^2 & r_k^{2^2} & \dots & r_k^{2^{k-1}} & r_k^{2^k} \\ x & x^2 & x^{2^2} & \dots & x^{2^{k-1}} & x^{2^k} \end{vmatrix}$$

In the expression for $D(x)$, if each r_i , $i = 1, 2, \dots, k$, is replaced by $r_i^{2^j}$, $j = 1, 2, \dots, t-1$, where $r_i^{2^t} = r_i$, then the corresponding coefficient of the LP representing C_0 is raised

to the power 2^j , $j = 1, 2, \dots, t-1$.

Q.E.D.

Illustration: Let us consider a linear (4,2) code consisting of code vectors r_0, r_1, r_2 and r_3 . Let r_0 be the zero code vector. Let r_1 and r_2 be a basis for this code. Thus $r_3 = r_1 + r_2$. Then

$$D_2 = \begin{vmatrix} r_1 & r_1^2 \\ r_2 & r_2^2 \end{vmatrix}$$

$$= r_1 r_2 (r_1 + r_2)$$

and $S_0(x)$ may be obtained as

$$S_0(x) = D(x)/D_2 = x^4 + a_1 x^2 + a_0 x$$

where $a_1 = (r_1^2 + r_2^2 + r_1 r_2)$ and $a_0 = (r_1 r_2 (r_1 + r_2))$

Substituting r_1 and r_2 by r_1^2 and r_2^2 respectively, we get the new coefficients, say a'_1 as $a'_1 = r_1^2 + r_2^2 + r_1^2 r_2^2 = a_1^2$, and a'_0 as $a'_0 = r_1^2 r_2^2 (r_1^2 + r_2^2) = a_0^2$.

Thus $S_1(x) = x^4 + a_1^2 x^2 + a_0^2 x$ and so on.

Example 6.4.1: In this example, we list in the form of a table (Table 6.3), all monic LPs of the form (6.1.1) with $a_0 \neq 0$, which divide $x^{16} - x$. Each of them represents the SP of a linear (4,2) code. The number of such polynomials is given by

$$N_{\text{dist}} = N_n / N_k = (2^4 - 1) \cdot (2^4 - 2) / (2^2 - 1) \cdot (2^2 - 2) = 35.$$

We assume that the polynomial coefficients belong to a NB of $GF(2^4)$. Let the primitive polynomial for generating $GF(2^4)$ be $x^4 + x + 1$. Let the primitive elements of $GF(2^4)$ in standard basis (SB) and in NB be β and σ respectively. We choose the normal basis vectors for $GF(2^4)$ as $\{\beta^3, \beta^6, \beta^{12}, \beta^9\}$. Thus for any $\sigma^u \in GF(2^4)$,

$$\sigma^u = m_{j3} \beta^9 + m_{j2} \beta^{12} + m_{j1} \beta^6 + m_{j0} \beta^3, \quad u = -\infty, 0, 1, \dots, 14,$$

where $\sigma^{-\infty} = 0000$, and $m_{ji} \in GF(2)$. All the 35 codes are put into different classes. Each class contains codes whose SP representations are of the form

$$S_j(x) = x^4 + (a_1)^{2^j} x^2 + (a_0)^{2^j} x,$$

where $j = 0, 1, 2, \dots, t-1$, $(a_i)^{2^t} = a_i$, $i = 0, 1$, $a_0 \neq 0$, and thus have the same weight distribution. The coefficients a_i are represented as a power of σ , the exponents of which are listed in the respective column, in the order $a_2 a_1 a_0$, where $a_2 = \sigma^0$. This is followed by the code vectors, again represented as a power of σ , the exponents of which are listed. In the last column is listed the cartesian representation of the code vectors in NB in the order $m_{j3} m_{j2} m_{j1} m_{j0}$.

Table 6.3: Normal Basis Syndrome Polynomials of Linear (4,2) Codes
Grouped into Classes on the basis of Same Weight Distributions

Class No.	Sl No.	Syndrome Polynomial			Code vectors in polar form in normal basis	Code vectors in cartesian form in normal basis			
		a_2	a_1	a_0					
1	1	0	14	11	$-\infty$ 2 3 6	0000	0011	0001	0010
	2	0	13	7	$-\infty$ 4 6 12	0000	0110	0010	0100
	3	0	11	14	$-\infty$ 8 12 9	0000	1100	0100	1000
	4	0	7	13	$-\infty$ 1 9 3	0000	1001	1000	0001
2	5	0	10	10	$-\infty$ 3 10 12	0000	0001	0101	0100
	6	0	5	5	$-\infty$ 6 5 9	0000	0010	1010	1000
3	7	0	1	14	$-\infty$ 3 4 7	0000	0001	0110	0111
	8	0	2	13	$-\infty$ 6 8 14	0000	0010	1100	1110
	9	0	4	11	$-\infty$ 12 1 13	0000	0100	1001	1101
	10	0	8	7	$-\infty$ 9 2 11	0000	1000	0011	1011

Table 6.3 (continued)

Class No	Sl No	Syndrome Polynomial			Code vectors in polar form in normal basis	Code vectors in cartesian form in normal basis			
		a_2	a_1	a_0					
4	11	0	11	4	∞ 3 5 11	0000	0001	1010	1011
	12	0	7	8	∞ 6 10 7	0000	0010	0101	0111
	13	0	14	1	∞ 12 5 14	0000	0100	1010	1110
	14	0	13	2	∞ 9 10 13	0000	1000	0101	1101
5	15	0	∞	9	∞ 3 8 13	0000	0001	1100	1101
	16	0	∞	3	∞ 6 1 11	0000	0010	1001	1011
	17	0	∞	6	∞ 12 2 7	0000	0100	0011	0111
	18	0	∞	12	∞ 9 4 14	0000	1000	0110	1110
6	19	0	8	2	∞ 0 3 14	0000	1111	0001	1110
	20	0	1	4	∞ 0 6 13	0000	1111	0010	1101
	21	0	2	8	∞ 0 12 11	0000	1111	0100	1011
	22	0	4	1	∞ 0 9 7	0000	1111	1000	0111
7	23	0	9	1	∞ 2 4 10	0000	0011	0110	0101
	24	0	3	2	∞ 4 8 5	0000	0110	1100	1010
	25	0	6	4	∞ 8 1 10	0000	1100	1001	0101
	26	0	12	8	∞ 1 2 5	0000	1001	0011	1010
8	27	0	6	14	∞ 2 13 14	0000	0011	1101	1110
	28	0	12	13	∞ 4 11 13	0000	0110	1011	1101
	29	0	9	11	∞ 8 7 11	0000	1100	0111	1011
	30	0	3	7	∞ 1 14 7	0000	1001	1110	0111
9	31	0	5	10	∞ 0 2 8	0000	1111	0011	1100
	32	0	10	5	∞ 0 4 1	0000	1111	0110	1001
10	33	0	0	5	∞ 10 11 14	0000	0101	1011	1110
	34	0	0	10	∞ 5 7 13	0000	1010	0111	1101
11	35	0	∞	0	∞ 0 5 10	0000	1111	1010	0101

6.5 Normal Basis Syndrome Polynomial Representations of Quasi Cyclic Codes

A linear (n,k) quasi cyclic code is a k -dimensional subspace which contains code vectors which are closed under block cyclic shifts. We denote a linear (n,k) quasi cyclic code which are closed under t cyclic shifts ($t \geq 1$) as a t -cyclic code.

In this section, we characterize t -cyclic codes using SPs with respect to a NB. The following theorem gives a characterization of t -cyclic codes by SPs in NB.

Theorem 6.5.1 Any linear (n,k) t -cyclic code Q can be uniquely represented by a monic LP over $GF(2^t)$ of degree 2^k , of the form

$$S(x) = \prod_{\beta \in Q} (x - \beta) = x^{2^k} + q_{k-1} x^{2^{k-1}} + \dots + q_0 x, \quad (6.5.1)$$

where $GF(2^t)$ is a subfield of $GF(2^n)$, β is a code vector considered as an element in some NB of $GF(2^n)$ belonging to Q , and $q_0 \neq 0$.

Proof: A linear (n,k) quasi cyclic code Q which is closed under t cyclic shifts has the property that if $\beta \in GF(2^n)$ is a code vector belonging to Q , then the element of $GF(2^n)$ which is obtained by cyclic shifting the n -tuple components of β by t positions, also belongs to Q . Now, if β is considered as an element in some NB of $GF(2^n)$, then β^q also belongs to Q where $q = 2^t$, as cyclic shifting the n -tuple components of β by t positions amounts to raising β to the power 2^t , in NB. Thus code vectors in Q can be grouped into cycles of the form $\beta_j, \beta_j^q, \beta_j^{q^2}, \dots, \beta_j^{q^{m_j-1}}$ where $\beta_j^{q^{m_j}} = \beta_j$, β_j being a representative member of the j^{th} cycle and $q = 2^t$. Let us consider the polynomial f_j which has $\beta_j^{q^i}$, $i = 0, 1, \dots, m_j-1$, as its roots. f_j is obviously the minimal polynomial of β_j over $GF(q)$, and thus can have coefficients only from $GF(q)$. Now the SP representing this code is a product of

these μ_j 's and therefore can have coefficients only from $GF(q)$. Further, $S(x)$ is a LP since its roots are the code vectors and the coefficient $q_0 \neq 0$, since there are no multiple roots.

Q.E.D.

Example 6.5.1: As an example, we refer to Table 6.3, which lists all the linear (4,2) codes. Classes 2 and 10 of this table give 2-cyclic codes, and we may see that the coefficients of their normal basis syndrome polynomial (NB SP) representation belong to $GF(2^2)$.

Next theorem is a converse to Theorem 6.5.1:

Theorem 6.5.2 Any monic LP $S(x)$ in NB, of degree 2^k over $GF(2^t)$ of the form (6.1.1), with $q_0 \neq 0$, which divides $x^{2^n} - x$, represents a linear (n,k) t-cyclic code, where the code vectors are members of the root space of $S(x)$, considered as elements represented in some NB of $GF(2^n)$, and $GF(2^t)$ is a subfield of $GF(2^n)$.

Proof: We use the result that for any field $GF(q)$, $q = \text{prime power}$, $x^{q^m} - x$ factors into monic irreducible polynomials over $GF(q)$ whose degrees divide m . In our case, $m = n/t$ and $q = 2^t$. Since $S(x)$ divides $x^{2^n} - x$ and $q_0 \neq 0$, the roots of $S(x)$ form the code vectors of a linear (n,k) code. Now as $x^{(2^t)^{n/t}} - x$ can be expressed as a product of monic irreducible polynomials over $GF(2^t)$, and $S(x)$ has coefficients from $GF(2^t)$, $S(x)$ may be expressed as a product of monic irreducible polynomials over $GF(2^t)$. This means that the roots of $S(x)$ can be grouped into cycles, the j^{th} cycle containing the roots β_j^i , $i = 0, 1, \dots, m_j - 1$, $q = 2^t$. In other words, $S(x)$ has roots whose n-tuple components are closed under t cyclic shifts if the roots are considered as elements in some NB of $GF(2^n)$. Thus $S(x)$ represents a t-cyclic code.

Q.E.D.

Theorem 6.5.3: If the NB SP

$$S_0(x) = x^{2^k} + q_{k-1}x^{2^{k-1}} + \dots + q_0x, \quad (6.5.2)$$

represents a linear (n,k) t -cyclic code, where the coefficients q_i 's $\in GF(2^t)$, then the NB SPs given by

$$S_j(x) = x^{2^k} + (q_{k-1})^{2^j}x^{2^{k-1}} + (q_{k-2})^{2^j}x^{2^{k-2}} + \dots + (q_0)^{2^j}x, \quad (6.5.3)$$

where $j = 1, 2, \dots, t-1$, $(q_i)^{2^t} = q_i$, $i = 0, 1, \dots, k-1$, $q_0 \neq 0$, also represent linear (n,k) t -cyclic codes. Further, the t -cyclic codes represented by S_j , $j = 0, 1, \dots, t-1$, have the same weight distribution.

Proof: Let $S_0(x)$ represent the linear (n,k) t -cyclic code Q_0 , whose roots are code vectors considered as elements belonging to some NB of $GF(2^n)$. Now let us consider linear (n,k) codes whose code vectors are obtained by cyclic shifting each of the n -tuple code vectors of Q_0 , by j places where $j = 0, 1, 2, \dots, t-1$ (t cyclic shifts results in the code Q_0 itself). We proved in Theorem 6.4.1 that the set of codes Q_j , $j = 0, 1, \dots, t-1$, will have the same weight distribution, and further $S_j(x)$, $j = 0, 1, \dots, t-1$, represent these codes. Now we prove that if Q_0 is t -cyclic, then Q_j , $j = 1, 2, \dots, t-1$, are also t -cyclic codes, as follows:

Since Q_0 is a t -cyclic code, it contains code vectors of the form β_ℓ^q , $i = 0, 1, \dots, m_\ell-1$ where $q = 2^t$, and β_ℓ is a member of the ℓ^{th} cycle. Now since Q_j , $j = 1, 2, \dots, t-1$, are obtained by cyclic shifting the n -tuple code vectors of Q_0 represented in a NB of $GF(2^n)$, they contain code vectors of the form $(\beta_\ell^{2^j})^q$, $i = 0, 1, \dots, m_\ell-1$, $j = 1, 2, \dots, t-1$. In other words, Q_j , $j = 1, 2, \dots, t-1$, are also t -cyclic codes, represented by the NB SPs $S_j(x)$, $j = 1, 2, \dots, t-1$. Q.E.D.

Examples. In the examples given below, we list (6,3) quasi cyclic codes and their NB SP representations. The code vectors are listed as a power of a primitive element, say σ , in some NB of $GF(2^6)$. Only the exponents of σ are listed. The codes listed are grouped into classes, each class containing codes with the same weight distribution. The primitive polynomial used for generating $GF(2^6)$ is chosen as $x^6 + x + 1$.

It may be noted that a cyclic code which is closed under single cyclic shifts is also closed under t cyclic shifts ($t > 1$). However, they are not listed.

Example 6.5.2 2-cyclic codes

Here all the (6,3) 2-cyclic codes are represented by NB SPs of degree 8, i.e., $S(x) = q_0x + q_1x^2 + q_2x^4 + q_3x^8$, with coefficients q_i from $GF(2^2)$ which is a subfield of $GF(2^6)$. $q_3 = 1$, since the polynomial is monic. The coefficients of these 2-cyclic codes will be from the field $GF(2^2)$ given by $\{\sigma^i\}$, where $i = -\infty, 0, 21$ and 42 .

Table 6.4: Normal Basis Syndrome Polynomials of (6,3) 2-Cyclic Codes
Grouped into Classes on the basis of Same Weight Distributions

Sl No	Syndrome Polynomial Coefficients				Code vectors in normal basis							
	q_0	q_1	q_2	q_3								
1	0	21	21	0	$-\infty$	0	13	14	19	35	52	56
	0	42	42	0	$-\infty$	0	26	28	38	7	41	49
2	0	21	42	0	$-\infty$	3	12	42	43	46	48	58
	0	42	21	0	$-\infty$	6	24	21	23	29	33	53
3	21	$-\infty$	$-\infty$	0	$-\infty$	3	12	21	30	39	48	57
	42	$-\infty$	$-\infty$	0	$-\infty$	6	24	42	60	15	33	51
4	21	21	0	0	$-\infty$	0	1	4	6	16	24	33
	42	42	0	0	$-\infty$	0	2	8	12	32	48	3

Table 6.5 (continued)

No	SP Coefficients				Code vectors in normal basis							
	q_0	q_1	q_2	q_3								
3	9	18	27	0	— ∞	5	9	18	29	40	43	54
	18	36	54	0	— ∞	10	18	36	58	17	23	45
	36	9	45	0	— ∞	20	36	9	53	34	46	27
4	9	27	54	0	— ∞	0	9	20	31	34	45	59
	18	54	45	0	— ∞	0	18	40	62	5	27	55
	36	45	27	0	— ∞	0	36	17	61	10	54	47
5	9	36	18	0	— ∞	0	11	22	25	36	50	54
	18	9	36	0	— ∞	0	22	44	50	9	37	45
	36	18	9	0	— ∞	0	44	25	37	18	11	27
6	9	45	45	0	— ∞	2	13	16	27	41	45	54
	18	27	27	0	— ∞	4	26	32	54	19	27	45
	36	54	54	0	— ∞	8	52	1	45	38	54	27
7	9	54	9	0	— ∞	4	7	18	32	36	45	56
	18	45	18	0	— ∞	8	14	36	1	9	27	49
	36	27	36	0	— ∞	16	28	9	2	18	54	35
8	27	0	27	0	— ∞	0	9	11	21	25	42	45
	54	0	54	0	— ∞	0	18	22	42	50	21	27
	45	0	45	0	— ∞	0	36	44	21	37	42	54
9	27	9	54	0	— ∞	0	2	12	16	33	36	54
	54	18	45	0	— ∞	0	4	24	32	3	9	45
	45	36	27	0	— ∞	0	8	48	1	6	18	27
10	27	18	18	0	— ∞	3	7	24	27	45	54	56
	54	36	36	0	— ∞	6	14	48	54	27	45	49
	45	9	9	0	— ∞	12	28	33	45	54	27	35
11	27	27	45	0	— ∞	15	18	36	45	47	57	61
	54	54	27	0	— ∞	30	36	9	27	31	51	59
	45	45	54	0	— ∞	60	9	18	54	62	39	55
12	27	36	9	0	— ∞	6	9	27	36	38	48	52
	54	9	18	0	— ∞	12	18	54	9	13	33	41
	45	18	36	0	— ∞	24	36	45	18	26	3	19
13	27	45	36	0	— ∞	0	18	27	29	39	43	60
	54	27	9	0	— ∞	0	36	54	58	15	23	57
	45	54	18	0	— ∞	0	9	45	53	30	46	51
14	27	54	0	0	— ∞	9	18	20	30	34	51	54
	54	45	0	0	— ∞	18	36	40	60	5	39	45
	45	27	0	0	— ∞	36	9	17	57	10	15	27

6.6 Normal Basis Syndrome Polynomial Representations of Cyclic Codes

The NB SPs of cyclic codes (t -cyclic codes where $t = 1$), have other interesting properties when compared to those representing t -cyclic codes where $t > 1$, and therefore deserve special attention. Therefore, in this section, we study the structure of NB SPs, whose roots form cyclic subspaces representing linear (n,k) cyclic codes.

Since a t -cyclic code can be represented by a NB SP whose coefficients belong to the subfield $GF(2^t)$ of $GF(2^n)$, and since cyclic codes can also be regarded as t -cyclic codes with $t = 1$, it is logical to assume that its NB SP will have coefficients from the ground field. However, we will prove this result differently to emphasize the additional structure of a cyclic subspace in NB.

6.6.1 Representation of a Cyclic Code as the Root Space of a Normal Basis P-Polynomial

If we recognize the elements of a cyclic subspace as elements in some NB of $GF(2^n)$, we immediately see that the same has the structure of a *modulus* M . According to Theorem A.3.8, a LP whose roots form a modulus has coefficients from the ground field. Thus we have the following important result:

Theorem 6.6.1: Any linear (n,k) cyclic code over $GF(p)$ can be represented by a p -polynomial of degree p^k of the form

$$G(x) = \prod_{\beta \in M} (x - \beta) = x^{p^k} + g_{k-1} x^{p^{k-1}} + \dots + g_0 x, \quad (6.6.1)$$

if the code vectors are recognized as elements in some NB of $GF(p^n)$, where $g_0 \neq 0$

and g_i 's $\in GF(p)$.

Proof: We limit our discussion to $p = 2$. By Theorem 6.1.1, a linear (n,k) cyclic code also can be represented by a monic LP over $GF(2^n)$ of degree 2^k . Now, it is required to prove that the coefficients are restricted to the ground field $GF(2)$. Any linear (n,k) cyclic code has a basis consisting of the generator polynomial $g(x)$, $x g(x)$, $x^2 g(x)$, ..., $x^{k-1} g(x)$ modulo $x^n - 1$, where $x^i g(x)$ modulo $x^n - 1$, $i = 1, 2, \dots, k-1$, amounts to cyclic shifting the code vector representing $g(x)$. Now, if we consider the code vector corresponding to $g(x)$ as an element y in some NB of $GF(2^n)$, then a basis of the cyclic code consists of k linearly independent vectors of the form $y, y^2, y^{2^2}, \dots, y^{2^{k-1}}$, since cyclic shifting the n tuples of y by j places corresponds to raising y to the 2^j th power, in NB. Thus the cyclic subspace consists of elements of the form $y, y^2, y^{2^2}, \dots, y^{2^{k-1}}$, and their linear combinations. This subspace is thus having the structure of a modulus. We have, from Theorem A.3.8, that the LP whose roots form a modulus, is a p -polynomial. Further, $g_0 \neq 0$, as in the case of any linear (n,k) code. Thus any linear (n,k) cyclic code can be represented by a NB p -polynomial of degree 2^k with $g_0 \neq 0$. Q.E.D.

We state a converse to the above theorem:

Theorem 6.6.2: Any p -polynomial of degree p^k of the form

$$G(x) = \prod_{\beta \in M} (x - \beta) = x^{p^k} + g_{k-1} x^{p^{k-1}} + \dots + g_0 x, \quad (6.6.2)$$

with $g_0 \neq 0$ and $g_i \in GF(p)$, which divides $x^{p^n} - x$, represents a linear (n,k) cyclic code if its code vectors recognized as elements with respect to an appropriate NB are considered as roots of the same.

Proof: We limit our discussion to $p = 2$ as before. Now since $G(x)$ is a LP of degree 2^k with $g_0 \neq 0$, and it divides $x^{2^n} - x$, the roots of the same represent a linear (n,k) code according to Theorem 6.1.2. To prove that this code is cyclic, we note the fact that $G(x)$ is

a p -polynomial whose roots satisfy the property that the p^{th} power of a root is again a root, i.e., the roots have the structure of a modulus consisting of a union of sets of the form $\{y, y^p, y^{p^2}, \dots\}$. This is a cyclic subspace if we consider the roots as belonging to an appropriate NB. Thus any $G(x)$ of the form given in (6.6.2) represents a cyclic code.

Q.E.D.

6.6.2 Computation of the Normal Basis P -Polynomials Representing a Given (n,k) Cyclic Code and its Dual $(n,n-k)$ Cyclic Code

Unlike the case of determination of SP representation of a general linear (n,k) code, the NB p -polynomial representation of a given linear (n,k) cyclic code can be found relatively easily. This follows from the results available on the class of p -polynomials (outlined in Appendix A). We proceed as follows:

Let $g(x)$ be the generator polynomial of the given linear (n,k) cyclic code and let $h(x)$ be its parity check polynomial. Then $g(x)$ and $h(x)$ both divide $x^n - 1$, and generate the cyclic code C and its dual code C_d (which is also cyclic) respectively. Further, they satisfy the relation

$$x^n - 1 = g(x) \cdot h(x). \quad (6.6.3)$$

We connect this relation, with the p -polynomials representing C and C_d using the notion of q -associates. We assume that $G(x)$ is the linearized q -associate of $h(x)$ and $H(x)$ is the linearized q -associate of $g(x)$. Then $G(x)$ and $H(x)$ satisfy the relation

$$x^{2^n} - x = H(x) (x) G(x) = G(x) (x) H(x), \quad (6.6.4)$$

where (x) denotes symbolic multiplication.

This is obtained by converting (6.6.3) to its linearized q -associate form. From the theory of p -polynomials, it follows that $H(x)$ is the dual LP of $G(x)$ and vice versa. Since $g(x)$ is of degree $n-k$, and $h(x)$ is of degree k , their linearized q -associates, namely $H(x)$

and $G(x)$ have degrees 2^{n-k} and 2^k respectively. The NB p -polynomial representing C , then must be $G(x)$ which is the linearized q -associate of $x^n-1/g(x) = h(x)$, and the NB p -polynomial representing C_d must be $H(x)$ which is the linearized q -associate of $x^n-1/h(x) = g(x)$.

Note: Because of the q -associate relationship, the NB SP representing the dual code of a cyclic code coincides with the corresponding dual LP. However, this is not so, in the case of any general linear code.

Example 6.6.1: Let us illustrate the case with the NB p -polynomial representations of a (7,3) cyclic code and its dual, namely a (7,4) cyclic code. First, we factorize x^7-1 into irreducible factors over $GF(2)[x]$ as

$$x^7-1 = (x+1)(x^3+x+1)(x^3+x^2+1).$$

$$\text{Let } g(x) = (x+1)(x^3+x+1) = x^4+x^3+x^2+1.$$

Then $h(x) = (x^3+x^2+1)$ generates the dual (7,4) cyclic code.

$$\text{Thus } x^7-1 = (x^4+x^3+x^2+1)(x^3+x^2+1)$$

Turning to their linearized q -associates, we get the relation

$$x^{2^7} - x = G(x) (x) H(x),$$

where $G(x) = x^8 + x^4 + x$ is the linearized q -associate of $x^3 + x^2 + 1 = x^{2^3} + x^{2^2} + 1$

The roots of this polynomial are the code vectors of the (7,3) cyclic code whose generator polynomial is $x^4 + x^3 + x^2 + 1$.

Let us determine the roots of $G(x)$ in a NB of $GF(2^7)$. We choose the primitive polynomial for generating $GF(2^7)$ as $x^7 + x^5 + x^2 + x + 1$. Let β and σ respectively be primitive elements of this field in SB and in NB. The roots of $G(x) = x^8 + x^4 + x$ in $GF(2^7)$ may be found as $0, \sigma^3, \sigma^6, \sigma^{12}, \sigma^{24}, \sigma^{48}, \sigma^{96}$ and σ^{65} . This set of roots expressed in any of the NBs of $GF(2^7)$ as polynomials in Θ^i , $i = 0, 1, \dots, 6$, where Θ^i is a normal basis of

$GF(2^7)$, gives the (7,3) cyclic code, the coefficients of the polynomials giving the n-tuple code vectors. We choose a NB, Θ^{2^i} , for $GF(2^7)$ as $\{\beta^5, \beta^{10}, \beta^{20}, \beta^{40}, \beta^{80}, \beta^{33}, \beta^{66}\}$. Thus any $\sigma^j = m_{j6} \beta^{66} + m_{j5} \beta^{33} + m_{j4} \beta^{80} + m_{j3} \beta^{40} + m_{j2} \beta^{20} + m_{j1} \beta^{10} + m_{j0} \beta^5$, where $m_{ji} \in GF(2)$.

The roots expressed in terms of m_{ji} 's are given in Table 6.6, where the m_{ji} 's are listed in the order $m_{j6} m_{j5} \dots m_{j0}$.

Table 6.6: Representation of a (7,3) Cyclic Code in Normal Basis

$\sigma^{-\infty}$	=	0 0 0 0 0 0 0
σ^3	=	1 0 1 0 0 1 1
σ^6	=	0 1 0 0 1 1 1
σ^{12}	=	1 0 0 1 1 1 0
σ^{24}	=	0 0 1 1 1 0 1
σ^{48}	=	0 1 1 1 0 1 0
σ^{96}	=	1 1 1 0 1 0 0
σ^{65}	=	1 1 0 1 0 0 1

Now let us determine the NB p-polynomial representation, $H(x)$, of the dual of this code, ie., a (7,4) cyclic code generated by the parity check polynomial $h(x)$ of the (7,3) cyclic code. This polynomial may be easily seen to be the linearized q-associate of $g(x) = x^4 + x^3 + x^2 + 1$. Thus $H(x) = x^{16} + x^8 + x^4 + x$. Using the same primitive polynomial and NB for $GF(2^7)$ as in the case of the (7,3) cyclic code considered in this example, the roots of $H(x)$ in $GF(2^7)$ may be found as 0, σ^0 , σ^{19} , σ^{38} , σ^{76} , σ^{25} , σ^{50} , σ^{100} , σ^{73} , σ^{29} , σ^{58} , σ^{116} , σ^{105} , σ^{83} , σ^{39} and σ^{78} . The roots expressed in terms of m_{ji} 's are given in Table 6.7, with the m_{ji} 's in the order $m_{j6} m_{j5} \dots m_{j0}$.

Table 6.7: Representation of a (7,4) Cyclic Code (which is the Dual of the (7,3) Cyclic Code given in Table 6.6) in Normal Basis

$\sigma^{-\infty}$	=	0 0 0 0 0 0 0
σ^0	=	1 1 1 1 1 1 1
σ^{19}	=	0 1 0 0 0 1 1
σ^{38}	=	1 0 0 0 1 1 0
σ^{76}	=	0 0 0 1 1 0 1
σ^{25}	=	0 0 1 1 0 1 0
σ^{50}	=	0 1 1 0 1 0 0
σ^{100}	=	1 1 0 1 0 0 0
σ^{73}	=	1 0 1 0 0 0 1
σ^{29}	=	1 1 1 0 0 1 0
σ^{58}	=	1 1 0 0 1 0 1
σ^{116}	=	1 0 0 1 0 1 1
σ^{105}	=	0 0 1 0 1 1 1
σ^{83}	=	0 1 0 1 1 1 0
σ^{39}	=	1 0 1 1 1 0 0
σ^{78}	=	0 1 1 1 0 0 1

This gives the (7,4) cyclic code whose generator polynomial is the parity check polynomial of the (7,3) cyclic code and whose NB p-polynomial representation is given by $H(x)$.

Theorem 6.6.3: Any p-polynomial $G(x)$ of degree 2^k of the form

$$G(x) = x^{2^k} + g_{k-1} x^{2^{k-1}} + \dots + g_0 x, \text{ with } g_0 \neq 0,$$

which divides $x^{2^n} - x$, represents a linear (n,k) cyclic code, and the corresponding dual p-polynomial of degree 2^{n-k} which divides $x^{2^n} - x$ represents its dual (n,n-k) cyclic code, the roots of both the polynomials in some NB of $GF(2^n)$ representing the respective cyclic subspaces.

Proof: Since $G(x)$ divides $x^{2^n} - x$ in the ordinary sense, it also divides $x^{2^n} - x$ symbolically, according to Theorem A.3.6. Thus it is possible to write

$$x^{2^n} - x = G(x) H(x),$$

for some p -polynomial $H(x)$ of degree $2^n - k$.

Turning to their conventional q -associates, we have

$$x^n - 1 = h(x) \cdot g(x)$$

where $h(x)$ is of degree k and $g(x)$ is of degree $n-k$. Thus $g(x)$ is a divisor of $x^n - 1$, of degree $n-k$, and represents the generator polynomial of a linear (n, k) cyclic code. Similarly $h(x)$ is a divisor of $x^n - 1$ and is of degree k , which is the generator polynomial of a $(n, n-k)$ cyclic code this being the dual of the (n, k) cyclic code generated by $g(x)$.

Q.E.D.

6.7 Study of Weight Distributions in Cyclic Codes

In a cyclic code, the code vectors can be grouped into cycles. The code vectors in each cycle are closed under cyclic shifts. Further, each cycle has code vectors of the same weight. Thus determination of weight distribution of a cyclic code reduces to the problem of identifying the cycles in the code. In this section, we show that the NB p -polynomial representation of a cyclic code, say $G(x)$, (where the code vectors are considered as elements represented in some NB of $GF(2^n)$) facilitates in finding these cycles, and hence the weight distribution. We formulate a new method for finding the weight distribution in a cyclic code from their NB p -polynomial representations, and illustrate it with suitable examples of well known block codes.

6.7.1 Determination of the Weight Distribution of Cyclic Codes from their Normal Basis P-Polynomial Representations

When the code vectors of a linear (n,k) cyclic code are considered as elements represented in a NB, a cycle in the code will have code vectors of the form $y, y^2, y^{2^2}, \dots, y^{2^{t-1}}$, where t is the length of the cycle. The minimal polynomial of y of degree t divides the p -polynomial $G(x)$ representing the cyclic code. Thus the factorization of $G(x)$ into irreducible polynomials over $GF(2)[x]$, helps in identifying the cycles in a cyclic code. The number of cycles is equal to the number of irreducible polynomials in the factorization. The number of members in each cycle is equal to the degree of each irreducible polynomial in the factorization. The weight distribution of the given cyclic code can be found by determining the weight of one representative root of each factor. The order of the representative root of each irreducible factor of degree t must be of order t .

We have developed a method for the factorization of polynomials over finite fields using the concept of DFT over finite fields. This is given in Appendix B. This is essentially a root finding algorithm and is particularly efficient if there are no repetitive roots and further, if the field in which the roots lie are known. Thus factorization of SPs can be efficiently done by this method.

We illustrate the method of determination of weight distribution of cyclic codes with some examples. In all the examples, we assign β and σ as primitive elements of $GF(2^n)$ in SB and in NB respectively. We denote the number of code vectors of weight i as A_i . Since x is always a factor of $G(x)$, the root $x = 0$, which corresponds to the all zero code vector, is always present, and $A_0 = 1$. Therefore, we list only the non-trivial cycles in the following examples:

Example 6.7.1: (i) We choose the (7,3) cyclic code given in Example 6.6.1 with

$g(x) = x^4 + x^3 + x^2 + 1$ and $h(x) = x^3 + x^2 + 1$. Let the primitive polynomial for generating $GF(2^7)$ be $x^7 + x^5 + x^2 + x + 1$. Let the NB for $GF(2^7)$ be chosen as $\{\beta^5, \beta^{10}, \beta^{20}, \beta^{40}, \beta^{80}, \beta^{33}, \beta^{66}\}$. $G(x)$ has been obtained as $G(x) = x^8 + x^4 + x$. Therefore $G(x)/x = x^7 + x^3 + 1$ which is an irreducible polynomial over $GF(2)[x]$. Thus the number of non-trivial cycles in this case is equal to 1 and its length is equal to 7. One root of $G(x)/x$ in NB is given by $\sigma^3 = 1 0 1 0 0 1 1$ whose weight is 4. Thus the weight distribution of this cyclic code is $A_0 = 1, A_4 = 7$.

(ii) Let us now compute the weight distribution of the dual of the above cyclic code i.e., a (7,4) cyclic code (also given in Example 6.6.1) whose $g(x) = x^3 + x^2 + 1$ and $h(x) = x^4 + x^3 + x^2 + 1$.

$$G(x) = x^{16} + x^8 + x^4 + x.$$

The q-associate relationship between $G(x)$ and $h(x)$ helps in the factorization of $G(x)$. Since $h(x)$ may be factorized into irreducible factors as $h(x) = (x+1)(x^3+x+1)$, $G(x)$ may be symbolically factorized into symbolically irreducible factors as $G(x) = (x^2+x)(x)(x^8+x^2+x)$. Since, each of these factors divides $G(x)$ symbolically, they divide $G(x)$ in the ordinary sense also. Thus (x^2+x) and (x^8+x^2+x) are factors of $G(x)$, or $(x+1)$ and (x^7+x+1) are irreducible factors of $G(x)/x$. The remaining irreducible factor can be found to be equal to $x^7+x^6+x^5+x^4+x^3+x^2+1$. Thus the number of non-trivial cycles in this case is equal to 3, with cycle lengths 1, 7 and 7 respectively. The representative roots are

- (1) $\sigma^0 = 1 1 1 1 1 1 1$ (root of $x+1$)
- (2) $\sigma^{29} = 1 1 1 0 0 1 0$ (a root of x^7+x+1)
- (3) $\sigma^{19} = 0 1 0 0 0 1 1$ (a root of $x^7+x^6+x^5+x^4+x^3+x^2+1$)

Thus the weight distribution of this cyclic code is given by

$$A_0 = 1, A_3 = 7, A_4 = 7, A_7 = 1.$$

Example 6.7.2: Let $n = 9$, $k = 3$. Let the primitive polynomial for generating $GF(2^9)$ be $x^9 + x^4 + 1$. Let the NB for $GF(2^9)$ be chosen as $\{\beta^5, \beta^{10}, \beta^{20}, \beta^{40}, \beta^{80}, \beta^{160}, \beta^{320}, \beta^{129}, \beta^{258}\}$.

$$x^9 - 1 = (x + 1)(x^2 + x + 1)(x^6 + x^3 + 1).$$

$$(i) \quad g(x) = x^6 + x^3 + 1. \text{ Then } h(x) = (x + 1)(x^2 + x + 1) = (x^3 + 1).$$

$$G(x) = \text{linearized } q\text{-associate of } h(x) = x^8 + x.$$

$$G(x)/x = (x + 1)(x^3 + x + 1)(x^3 + x^2 + 1).$$

Thus there are 3 non-trivial cycles of length 1, 3 and 3 respectively. The representative roots are

$$(1) \quad \sigma^0 = 111111111 \text{ (root of } x + 1)$$

$$(2) \quad \sigma^{73} = 011011011 \text{ (a root of } x^3 + x + 1)$$

$$(3) \quad \sigma^{219} = 100100100 \text{ (a root of } x^3 + x^2 + 1)$$

Thus the weight distribution of this cyclic code is

$$A_0 = 1, A_3 = 3, A_6 = 3 \text{ and } A_9 = 1.$$

(ii) Let us take the dual of the (9,3) cyclic code in (i) i.e., a (9,6) cyclic code, and find its weight distribution by the factorization of its NB p -polynomial representation. Let the primitive polynomial for $GF(2^9)$ and the NB be same as in (i). Then $g(x) = x^3 + 1$ and $h(x) = x^6 + x^3 + 1$. Therefore $G(x) = \text{linearized } q\text{-associate of } h(x) = x^{64} + x^8 + x$. $G(x)/x = x^{63} + x^7 + 1$.

Since $h(x)$ is irreducible in $GF(2)[x]$, the degree of every irreducible factor of $G(x)/x$ is equal to the order of $h(x)$, according to Theorem A.3.7. The order of $h(x)$ may be found to be equal to 9. Thus $G(x)/x$ can be factorized into 7 irreducible polynomials each of degree 9. Thus there are 7 non-trivial cycles each of length 9. These irreducible polynomials and their representative roots are listed in Table 6.8. Only the exponents of x

polynomials, they are not listed. Thus an entry '9 6 5 2' would mean the irreducible polynomial $x^9 + x^6 + x^5 + x^2 + 1$. The roots are listed as a power of σ , the exponents of which are listed in the respective column.

Table 6.8: Irreducible Polynomials in the Factorization of $x^{63} + x^7 + 1$ and their Representative Roots in Normal Basis

Sl No.	Irreducible Polynomials in the factorization of $x^{63} + x^7 + 1$	Representative roots in normal basis	
		Polar	Cartesian
1	9 7 4 2	53	101011110
2	9 7 5 1	107	010110100
3	9 6 5 2	63	010011001
4	9 5 4 1	253	110111001
5	9 7 6 4 3 1	17	110000110
6	9 6 5 4 3 2	45	111000111
7	9 7 6 3 2 1	59	010000010

Thus the weight distribution of this cyclic code is

$$A_0 = 1, A_2 = 9, A_4 = 3 \times 9 = 27 \text{ and } A_6 = 3 \times 9 = 27.$$

Example 6.7.3: Let $n = 4$, $k = 3$. Let the primitive polynomial for generating $GF(2^4)$ be $x^4 + x + 1$. Let the NB for $GF(2^4)$ be chosen as $\{\beta^3, \beta^6, \beta^{12}, \beta^9\}$.

$$x^4 - 1 = g(x) \cdot h(x)$$

where $g(x) = x + 1$, and $h(x) = x^3 + x^2 + x + 1$.

$G(x)$ = linearized q -associate of $h(x) = x^8 + x^4 + x^2 + x$.

$$G(x)/x = (x + 1)(x^2 + x + 1)(x^4 + x + 1).$$

Thus there are 3 nontrivial cycles of length 1, 2 and 4 respectively. The representative

roots are

- (1) $\sigma^0 = 1\ 1\ 1\ 1$ (root of $x + 1$)
- (2) $\sigma = 1\ 0\ 0\ 1$ (a root of $x^4 + x + 1$)
- (3) $\sigma^5 = 1\ 0\ 1\ 0$ (a root of $x^2 + x + 1$)

Thus the weight distribution of this cyclic code is

$$A_0 = 1, A_2 = 4 + 2 = 6 \text{ and } A_4 = 1.$$

Example 6.7.4: Let $n = 3$, $k = 2$. Let the primitive polynomial for generating $GF(2^4)$ be $x^3 + x^2 + 1$. Let the NB for $GF(2^4)$ be chosen as $\{\beta, \beta^2, \beta^4\}$.

$$x^3 - 1 = g(x) \cdot h(x).$$

where $g(x) = x + 1$, and $h(x) = x^2 + x + 1$.

$G(x) =$ linearized q -associate of $h(x) = x^4 + x^2 + x$.

$$G(x)/x = x^3 + x + 1.$$

Thus there is only one non-trivial cycle whose length is 3.

$\sigma^3 = 1\ 0\ 1$, is a root of $G(x)/x$.

Thus the weight distribution of this cyclic code is

$$A_0 = 1, A_2 = 3.$$

Next we give a few examples of SP representations of BCH codes. Since BCH codes are cyclic, they may also be represented by p -polynomials in NB.

6.7.2 Examples of Normal Basis P-Polynomial Representations of BCH Codes

We list the NB p -polynomial representation of 3 BCH codes of block length 15 along with their dual codes in the following examples. We find the weight distributions by factorizing the respective NB p -polynomials. In all the examples, we take the primitive

σ as a primitive element in NB. We choose the NB for $GF(2^{15})$ as $\{\beta^{29}, \beta^{58}, \beta^{116}, \beta^{232}, \beta^{464}, \beta^{928}, \beta^{1856}, \beta^{3712}, \beta^{7424}, \beta^{14848}, \beta^{29696}, \beta^{59392}, \beta^{118784}, \beta^{237568}, \beta^{475136}\}$.

In all the examples, we denote the generator polynomial as $g(x)$ and the parity check polynomial as $h(x)$. The NB p -polynomial representation of the respective cyclic code is denoted by $G(x)$ and that of the dual code is denoted by $H(x)$.

Example 6.7.5: (a) (15,7) BCH code

$$g(x) = x^8 + x^7 + x^5 + x^4 + x^3 + x + 1 = (x^4 + x + 1)(x^4 + x^3 + 1).$$

$$h(x) = x^7 + x^6 + x^5 + x^2 + x + 1.$$

$$= (x + 1)(x^2 + x + 1)(x^4 + x^3 + x^2 + x + 1).$$

$$G(x) = \text{linearized } q\text{-associate of } h(x) = x^{128} + x^{64} + x^{32} + x^4 + x^2 + x.$$

The nontrivial irreducible factors (other than x) of $G(x)$ are listed in Table 6.9 along with a representative root. The possible degrees of the irreducible factors are 1, 3, 5 and 15 corresponding to the possible cycle lengths which divide n . Only the exponents of x in the irreducible polynomials are listed in the second column. Further, the last term ' $+1$ ' which is present in every irreducible polynomial, is not listed. Thus a polynomial, say, $x^5 + x^2 + 1$, is listed as 5 2. Similarly, in the third column, only the exponents ' i ' of the representative root, σ^i , in NB, are listed.

It may be seen from the table that the weight distribution of this code is given by $A_0 = 1; A_3 = 5; A_5 = 3; A_6 = 25; A_7 = 30; A_8 = 30; A_9 = 25; A_{10} = 3; A_{12} = 5; A_{15} = 1$.

(b) The dual of the BCH code in (a) is the (15,8) cyclic code generated by the $h(x)$ of the same. The NB p -polynomial representation of this code, $H(x)$, is given by the linearized q -associate of $g(x)$. Thus $H(x)$ is given by

$$H(x) = x^{256} + x^{128} + x^{32} + x^{16} + x^8 + x^2 + x.$$

The nontrivial factors (other than x) consisting of 17 irreducible polynomials of degree 15, are listed in Table 6.10. There are no factors of degree 3 and 5.

Table 6.9 Irreducible Polynomials in the Factorization of $G(x)/x$ and their Representative Roots in Normal Basis

Sl No	Irreducible Polynomials in the factorization of $G(x)/x$	Representative Roots in normal basis	
		Polar form	Cartesian form
1	1	0	11111111111111
2	3 1	4681	011011011011011
3	3 2	14043	100100100100100
4	5 2	5285	101111011110111
5	5 3 2 1	1057	011000110001100
6	5 4 2 1	7399	000100001000010
7	5 4 3 1	3171	111001110011100
8	5 4 3 2	15855	011010110101101
9	5 3	11627	001010010100101
10	15 12 5 4 3 2	19	010011111100100
11	15 13 8 5 2 1	261	001011001101000
12	15 14 13 11 10 9 7 6 2 1	1263	101100000011011
13	15 14 11 10 8 7 6 5 4 3	1227	001100101111101
14	15 14 13 12 10 9 6 5 4 1	2485	111000101010001
15	15 11 10 9 7 6 5 4 3 1	3477	100001110101011

Table 6.10: Irreducible Polynomials in the Factorization of $H(x)/x$ and their Representative Roots in Normal Basis

Sl No	Irreducible Polynomials in the factorization of $H(x)/x$	Representative Roots in normal basis	
		Polar form	Cartesian form
1	15 12 11 8 5 4 2 1	55	010010000001001
2	15 13 8 7 6 3 2 1	1065	110111101100000
3	15 13 11 10 7 6 5 4 3 2	2413	000100001100001
4	15 12 11 7 6 5 2 1	2869	000011111011111
5	15 1	1	010111100010011
6	15 7	1335	011110101100100
7	15 13 11 10 9 8 7 6 5 4 3 2	703	000010101101010
8	15 13 12 9 8 6 5 4 3 1	2463	110111010101110

Table 6.10 (continued)

Sl No	Irreducible Polynomials in the factorization of $H(x)/x$	Representative Roots in normal basis	
		Polar form	Cartesian form
9	15 13 11 6 3 2	3235	011111111010001
10	15 11 10 9 8 6	4077	100000001110011
11	15 12 10 7 6 2	4983	100111011100100
12	15 13 12 10 8 6 4 2	5287	001010101001111
13	15 13 12 10 9 5 4 1	5563	111111001101100
14	15 13 12 11 9 8 7 4	7415	101101101001100
15	15 12 11 10 6 5 4 3	7839	000110011000101
16	15 11 10 9 3 1	11643	111010010111000
17	15 11 8 6 5 2	15327	011010000101100

From Table 6.10, the weight distribution of this code is given by $A_0 = 1$; $A_4 = 30$; $A_6 = 60$; $A_8 = 105$; $A_{10} = 60$.

Example 6.7.6: (a) (15,5) BCH code

$$g(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$$

$$= (x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$$

$$h(x) = x^5 + x^3 + x + 1 = (x + 1)(x^4 + x^3 + 1)$$

$$G(x) = \text{linearized } q\text{-associate of } h(x) = x^{32} + x^8 + x^2 + x$$

The irreducible factors of $G(x)/x$ are listed in Table 6.11.

The weight distribution of this code is given by

$$A_0 = 1; A_7 = 15; A_8 = 15; A_{15} = 1.$$

Table 6.11: Irreducible Polynomials in the Factorization of $G(x)/x$ and their Representative Roots in Normal Basis

Sl No	Irreducible Polynomials in the factorization of $G(x)/x$	Representative Roots in normal basis	
		Polar form	Cartesian form
1	1	0	111111111111111
2	15 7	24909	000111101011001
3	15 14 13 12 11 10 9 8	22239	000010100110111

(b) The dual of the BCH code in (a) is the (15,10) cyclic code generated by its $h(x)$.

The NB p -polynomial representation of this code, $H(x)$, is given by

$$H(x) = x^{1024} + x^{256} + x^{32} + x^{16} + x^4 + x^2 + x.$$

The nontrivial factors (other than x) consist of 17 irreducible polynomials of degree 15. There are no factors of degree 3 and 5.

Some of the irreducible factors of $H(x)/x$ which are directly derivable from $g(x)$, are listed as follows:

$$(1) \quad (\text{Linearized } q\text{-associate of } x^2 + x + 1)/x = x^3 + x + 1$$

$$(2) \quad (\text{Linearized } q\text{-associate of } x^4 + x + 1)/x = x^{15} + x + 1$$

and (the linearized q -associate of $x^4 + x^3 + x^2 + x + 1$)/ x having 3 irreducible factors of degree 5 (Since $x^4 + x^3 + x^2 + x + 1$ is irreducible in $GF(2)[x]$, the degree of every irreducible factor of its linearized q -associate/ x is equal to its order, which, in this case, is equal to 5):

$$(3) \quad x^5 + x^2 + 1$$

$$(4) \quad x^5 + x^3 + x^2 + x + 1$$

$$(5) \quad x^5 + x^3 + 1$$

The remaining irreducible factors are of degree 15, 66 in number.

Thus in this code, there are 1 cycle of length 1, 1 cycle of length 3, 3 cycles of length 5 and 67 cycles of length 15.

Example 6.7.7: (a) (15,11) BCH code

$$g(x) = (x^4 + x + 1)$$

$$\begin{aligned} h(x) &= x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1 \\ &= (x^4 + x^3 + x^2 + x + 1)(x^4 + x^3 + 1)(x^2 + x + 1)(x + 1) \end{aligned}$$

$$G(x) = x^{2048} + x^{256} + x^{128} + x^{32} + x^8 + x^4 + x^2 + x$$

Some of the irreducible factors of $G(x)/x$, which are directly derivable from $h(x)$, are listed as follows:

The (linearized q -associate of $x^4 + x^3 + x^2 + x + 1$)/ x having 3 irreducible factors of degree 5 as

- (1) $x^5 + x^2 + 1$
- (2) $x^5 + x^3 + x^2 + x + 1$
- (3) $x^5 + x^3 + 1$
- (4) (linearized q -associate of $x^4 + x^3 + 1$)/ $x = x^{15} + x^7 + 1$
- (5) (linearized q -associate of $x^2 + x + 1$)/ $x = x^3 + x + 1$
- (6) (linearized q -associate of $x + 1$)/ $x = x + 1$

Out of the remaining irreducible factors, there is one polynomial of degree 3, namely,

$$(7) \quad x^3 + x^2 + 1$$

and 3 polynomials of degree 5 namely,

- (8) $x^5 + x^4 + x^2 + x + 1$
- (9) $x^5 + x^4 + x^3 + x + 1$
- (10) $x^5 + x^4 + x^3 + x^2 + 1$

The remaining 133 irreducible factors are of degree 15.

Thus in this code, there are 2 cycles of length 1, 2 cycles of length 3, 6 cycles of length 5 and 134 cycles of length 15.

(b) The dual of the BCH code in (a) is the (15,4) cyclic code generated by its $h(x)$. The NB p-polynomial representation of this code, $H(x)$, is given by

$$H(x) = x^{16} + x^2 + x = x(x^{15} + x + 1).$$

$H(x)/x$ is known to be irreducible over $GF(2)$. One root of $x^{15} + x + 1$ is $\sigma = 010111100010011$. Thus the weight distribution of this code is $A_0 = 1, A_8 = 15$.

In the next subsection, we give the examples of NB p-polynomial representations of cyclic codes which are equivalent to the (23,12) Golay codes:

6.7.3 Examples of Normal Basis P-Polynomial Representations of Golay Codes

$x^{23} + 1$ may be factorized as

$$x^{23} + 1 = (x + 1) g_1(x) g_2(x), \text{ where}$$

$$g_1(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$$

and
$$g_2(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1.$$

The cyclic codes equivalent to the Golay code are generated by $g_1(x)$ and $g_2(x)$.

Example 6.7.8: (i) The parity check polynomial corresponding to $g_1(x)$, say $h_1(x)$, is given by

$$h_1(x) = (x + 1) g_2(x) = x^{12} + x^{10} + x^7 + x^4 + x^3 + x^2 + x + 1.$$

Thus the NB p-polynomial representation of this Golay code is given by

$$\begin{aligned} G_1(x) &= \text{linearized } q\text{-associate of } h_1(x) \\ &= x^{4096} + x^{1024} + x^{128} + x^{16} + x^8 + x^4 + x^2 + x. \end{aligned}$$

$G_1(x)/x$ has one irreducible factor of degree 1, namely $(x + 1)$, and the remaining irreducible factors are of degree 23, 178 in number.

(ii) The parity check polynomial corresponding to $g_2(x)$, say $h_2(x)$, is given by

$$h_2(x) = (x+1)g_1(x) = x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1.$$

Thus the NB p -polynomial representation of this Golay code is given by

$$\begin{aligned} G_2(x) &= \text{linearized } q\text{-associate of } h_2(x) \\ &= x^{4096} + x^{2048} + x^{1024} + x^{512} + x^{256} + x^{32} + x^4 + x. \end{aligned}$$

Similar to the first code, $G_2(x)/x$ also has one irreducible factor of degree 1, namely, $(x+1)$, and 178 irreducible factors of degree 23.

We conclude this chapter with a description of the NB-polynomial representations of self dual cyclic codes and their weight distributions.

6.7.4 Normal Basis P -Polynomial Representations of Self Dual Cyclic Codes

A binary self-dual code is a code whose dual code is the code itself. Such codes then must be of even block length with $n = 2k$. Some self-dual codes have the additional structure of being cyclic. We characterize these codes in this subsection by their NB p -polynomial representations.

Since these codes are cyclic as well as self dual, they have a generator polynomial of the form $x^k + 1$, since $h(x) = g(x)$ in their case and $n = 2k$.

Thus their NB p -polynomials are of the form

$$S(x) = x^{2^k} + x. \quad (6.7.1)$$

It is well known that this polynomial splits in $GF(2^k)$. Thus we get the following information from their NB p -polynomial representations:

- (1) Number of cycles in this code is equal to the number of Frobenius classes in $GF(2^k)$.
- (2) Number of elements in each cycle is equal to the order of the Frobenius class.

Example 6.7.9: For example, the linear (4,2) code listed in class 11 of Table 6.3 is a self dual (4,2) cyclic code with its NB p -polynomial being given by $x^4 + x$.

CHAPTER 7

CONCLUSIONS

We have studied the class of Galois switching functions (GSFs) with regard to their algebraic theory and structures, and their utility in representation, characterization, analysis and classification of discrete signals encountered in switching and coding theories by means of Galois polynomials (GPs). Our studies have been confined to GPs over finite fields of characteristic 2. Specifically, GSFs representing discrete signals characterized by the following mappings have been taken up for our study:

- (1) Mappings from $GF(2^k)$ to $GF(2^n)$ in general, with particular reference to $n = 1$, resulting in the class of Boolean functions (BFs) and their characterizations.
- (2) Linear mappings from $GF(2^k)$ to $GF(2^n)$ where k is not necessarily equal to n and the mapping generally many-to-one, described by linearized Galois polynomials (LGPs); they being called linearized Frobenius polynomials (LFPs) if their coefficients satisfy nontrivial conjugacy constraints and called simply linearized polynomials (LPs) if the conjugacy relations are trivial.
- (3) Linear mappings from $GF(2^k)$ to $GF(2^n)$ where $k < n$, and the mappings one-to-one, leading to linear (n,k) block codes and their representations by LGPs.
- (4) Specialized many-to-one linear mappings from $GF(2^n)$ to $GF(2^n)$ characterized by pairs of linearized polynomials (LPs) called syndrome polynomials (SPs), the k -dimensional root space of one LP constituting the range space of the other and vice versa, thus providing alternate characterization of block codes, other than (3), by means of SPs.

Some of the notable features of GSFs representing mappings of the kind described above and their GP representations are:

- (1) GSFs possess well defined algebraic properties thus opening up the possibility of the study of signals and systems described by them in terms of algebraic models. Due to this algebraic characterization, it is possible to conduct transform domain studies of those discrete signals whose lengths are not relatively prime to the characteristic of the finite field by means of DFT like finite field transforms.
- (2) GPs allow compact representation of discrete signals and systems. They provide a very effective means for the utilization of GSFs in various applications.
- (3) GP representations of GSFs exhibit remarkable properties connected with conjugacy relations if the range values belong to a proper subfield of the finite field to which the coefficients of the polynomial belong. This leads to implementational advantages by the use of normal basis (NB) representation for realization of discrete signals.
- (4) GSFs allow representations of linear transformations and linear block codes in terms of LGPs. Representations of block codes open up alternative methods for their encoding and decoding.
- (5) NB LP representations of GSFs have special significance to coding theory, particularly, in the characterization of well known block codes and in the study of their weight distributions.

7.1 Summary of Results

Results obtained in this thesis are summarized as follows:

GSFs represent mappings from $GF(2^k)$ to $GF(2^n)$ and are defined as discrete signals with finite index sets having the structure of a multiplicative cyclic monoid $M(2^k)$ and assuming values from a finite field $GF(2^n)$. The monoid structure of the index set has resulted in attributing the structure of a monoid algebra to the class of GSFs with pointwise addition and an appropriately defined convolution. Further, a DFT like finite

field transform called the Galois Transform (GT) is defined on these signals, thus enabling a finite field transform domain study of discrete signals whose lengths are not relatively prime to the characteristic of the field.

It is shown that the monoid algebra model of GSFs accommodates the existing representations of GSFs. Frobenius cycles among the coefficients of GPs representing GSFs are examined and it is shown that when $k \nmid n$, nontrivial conjugacy relations exist whereas when $k|n$, in general, the relations are trivial, even though in some cases, nontrivial Frobenius cycles show up. It is observed that the Frobenius cycles existing in GPs in fact correspond to members of minimal ideals in an appropriate monoid algebra. The advantage of NB representation in the computation of Frobenius sums is pointed out. Further, an existing expression for the number of conjugate cycles in the case of DFT is suitably modified to obtain the number of Frobenius cycles in the case of GPs representing GSFs.

Since the class of BFs constitutes a subclass of the general class of GSFs, the former is also characterized by an appropriate monoid algebraic structure. It is shown that any BF has a GP representation in the form of a Frobenius polynomial (FP) (ie., a polynomial consisting of Frobenius terms) out of which the standard class of linear Boolean functions (LBFs) has GP representations in the form of linearized Frobenius polynomials (LFPs), which are LGPs whose coefficients satisfy conjugacy constraints. A study of the ideal structures in the monoid algebra of BFs shows that the class of LBFs forms a minimal ideal in this algebra. Further, the class of generalized Reed-Muller (GRM) codes constructed from LBFs, also has the structure of an ideal in the appropriate monoid algebra, this being expressed by a direct sum of certain minimal ideals determined by the order and dimension of the code. GP representations of the basis vectors of a GRM code are determined in terms of those of LBFs and it is shown that an r^{th} order GRM code of block length 2^m , has at least $2^{m-r}-1$ consecutive GP coefficients equal to zeroes.

Classification problems in BFs are examined. The traditional equivalence relations used to classify BFs commonly known as the five invariance operations are studied in terms of their effect on the GP coefficients of the corresponding BFs. This study has enabled the modeling of a system which realizes BFs from a prototype function in a class of functions (classified using the five invariance operations), which entirely employs finite field modules. A class identification procedure for 2 and 3 variable BFs by verification of their GP coefficients is proposed. The suitability of certain operations connected with the monoid algebraic structure of BFs in the classification of the same, is then examined for 2 and 3 variable BFs. It is observed that the classes obtained by these relations contain elements from different ideals in the corresponding monoid algebra. A finite field model which realizes BFs as a sum of elements from minimal ideals in the monoid algebra is proposed. This turns out to be a Frobenius sum computer and the use of NB representation in its implementation is suggested.

The study of the five invariance operations and its effect on the GP coefficients has helped in characterizing β -self dual (SD)/anti self dual (ASD) BFs in the transform domain by means of their GP coefficients. Thus the constraints on the GP coefficients of any BF to be β -SD/ASD, are derived for 2, 3 and 4 variable cases.

A linear (n,k) transformation is defined as a linear transformation from the k -tuple vector space to the n -tuple vector space, where k is not necessarily equal to n . The possibility of a transform domain study of linear (n,k) transformations which includes the class of linear (n,k) block codes (those transformations which represent one-to-one mappings) is pointed out as another advantage of treating discrete signals and systems as members of a monoid algebra. Unlike DFT over finite fields, which can be defined only on code lengths that are relatively prime to the characteristic of the finite field, it is shown that the extended DFT defined in a monoid algebra allows all linear (n,k) transformations to be studied in the transform domain by treating them as discrete signals over multiplicative cyclic monoids $M(2^k)$ which assume values from $GF(2^n)$. The conditions for

a GSF to be linear are derived. The resulting functions which represent linear mappings from $GF(2^k)$ to $GF(2^n)$ are denoted as linearized GSFs (LGSFs) and their GP representations are called linearized Galois polynomials (LGPs). It is shown that there exists an isomorphism between the class of linear (n,k) transformations and the corresponding class of LGSFs. Further, the class of LGSFs is shown to exhibit the property of an ideal in a monoid algebra. It is shown that the LGP coefficients of a LGSF representing a linear (n,k) transformation is derivable from the matrix of n -tuple vectors which generates the transformation, and that they are related to these vectors by means of a Vander monde matrix assuming values from $GF(2^k)$. The standard basis (SB) and NB representations of this Vander monde matrix are then derived.

Conjugacy relations among the coefficients of LGPs representing LGSFs are studied and those functions whose LGP coefficients exhibit nontrivial conjugacy relations are shown to be represented by a LFP. Those functions whose LGP coefficients possess trivial conjugacy relations are denoted simply as linearized functions (LFs) and they are represented by LGPs denoted simply as linearized polynomials (LPs). Algebraic structures for LGPs are formulated under an operation commonly known as symbolic multiplication. Even though this operation is noncommutative, it is shown that by grouping the polynomials in a particular manner, the class of *single term LGPs* under both cases of trivial and nontrivial conjugacy relations possess the structure of a finite field isomorphic to $GF(2^n)$ under the binary operations of pointwise addition and symbolic multiplication.

Only those LGPs of a given pair of n and k satisfying certain nonzero determinant property among its GP coefficients are shown to represent linear (n,k) block codes, the rest of these functions for the same n and k representing many-to-one mappings. LGP representations of classes of linear block codes with the same weight distribution are derived in the NB representation. It is shown that by grouping LGPs in the same way as in the case of single term LGPs, it is possible to separate out LGPs representing one-to-one mappings (linear block codes) and those representing many-to-one mappings, provided the

nature of at least one of the mappings in each group is known. Further, the properties of the class of single term LGPs are investigated and it is first shown that when $k|n$, single term LGPs always represent one-to-one mappings. Secondly, a study of the distinctness of codes generated by single term LGPs which are members of a finite field, are undertaken. The number of distinct codes in each field is computed. It is shown that when n and k are relatively prime, all the codes in the corresponding finite field are distinct.

The roots of the LGPs representing linear block codes are examined and it is observed that they are not in general confined to the same field. Further, the possibility of characterizing individual codes by the roots of their LGPs is explored. It is found that roots characterize group of codes rather than individual ones. It is proved that the roots of a LGP representing a linear (n,k) block code cannot assume nonzero values from $GF(2^k)$.

Expressions for LGP representations of cyclic codes both in SB and in NB are derived. It is shown that it is possible to represent some cyclic codes whose $k|n$, by a p -polynomial, i.e., a LGP whose coefficients belong to the ground field.

Role of GSFs in the decoding of linear block codes is considered. A standard array for a linear (n,k) block code consists of a matrix of 2^{n-k} rows and 2^k columns containing $GF(2^n)$ vectors and therefore any standard array is shown to be representable by a 2-D GSF, the coefficients of its 2-variable GP obtained by computing the 2-D Galois transform (GT) of the matrix of $GF(2^n)$ vectors constituting the standard array. It is shown that the two-variable GP representing a standard array has, in general, nonzero coefficients only in the first row and in the first column of the matrix of the resultant 2-D GT coefficients, corresponding respectively to the single variable GP coefficients representing the linear block code (which is a LGP), and the coset leaders. It is shown that a variety of techniques are possible for the standard array decoding of a linear block code using both 1-D and 2-D GSFs depending on one's choice of the received vector to be decoded into an n -tuple code vector or a k -tuple message. It is shown that a decoder which implements standard array decoding of a received vector into a k -tuple message

using 1-D GSFs always has the form of a Frobenius sum computer. As a final result on the application of GSFs in coding, the GP representations of syndrome tables associated with a standard array are examined and it is shown that any GP which computes syndrome has a LFP representation. Further, those roots of this polynomial which belong to $GF(2^n)$, form the code vectors of the corresponding linear (n,k) block code.

Syndrome Polynomials (SPs) are defined as LPs over $GF(2^n)$ of degree 2^k which divide $x^{2^n} - x$ whose roots are nonrepetitive and form the code vectors of a linear (n,k) block code. They describe special types of GSFs representing many-to-one linear mappings from $GF(2^n)$ to $GF(2^n)$ described by a pair of LPs, the root space of one forming the range space of the other and vice versa. For the roots to be nonrepetitive, it is argued that this LP in x should have the coefficient of x nonzero. It is shown that any linear (n,k) block code has a SP representation and conversely, any LP in x over $GF(2^n)$ of degree 2^k with the coefficient of x nonzero, which divides $x^{2^n} - x$ represents a linear (n,k) block code as its root space. It is proved that, because of the existence of a dual LP for every SP which is of degree 2^{n-k} over $GF(2^n)$ with the coefficient of x nonzero and which also divides $x^{2^n} - x$, every SP can be used for computing syndromes for the code which it represents, the syndromes being members of the root space of the corresponding dual LP, they being n -tuples instead of the conventional $(n-k)$ tuple syndromes associated with a standard array.

It is pointed out that SPs in NB representation have a special significance in coding theory. To support this argument, it is first shown that it is possible to classify codes on the basis of same weight distribution by means of their NB SP representations. Secondly, the class of t -cyclic codes (quasi cyclic codes that are closed under t cyclic shifts where $t \geq 1$) are shown to be completely characterized by their NB SPs which has coefficients belonging to $GF(2^t)$, where $GF(2^t)$ is a subfield of $GF(2^n)$. Conversely, it is shown that any NB SP in x over $GF(2^t)$ with the coefficient of x nonzero, which divides $x^{2^n} - x$, represents

a linear (n,k) t -cyclic code. Further, t -cyclic codes of the same weight distribution are classified on the basis of their NB SP representations. From the result on the representability of t -cyclic codes by a SP over $GF(2^t)$, it follows that when $t = 1$, the coefficients belong to the ground field and hence the associated SP represents a cyclic code in the form of a NB p -polynomial. However, this result is proved from a different angle by taking note of the fact that a cyclic subspace has the structure of a modulus and that a p -polynomial is associated with every modulus, the latter forming the root space of the former. It is further shown that the NB p -polynomial representation of a cyclic code is easily derivable from the generator polynomial of its dual cyclic code and is equal to the linearized q -associate of the same. A new method to determine the weight distributions in cyclic codes is proposed. This is by factorization of the NB p -polynomial representations of the same. A procedure for factorization of polynomials over finite fields is also developed, which is a root finding algorithm and which makes use of the DFT over finite fields. Factorization of the NB p -polynomials representing cyclic codes gives the following information about their weight distributions:

- (1) The number of cycles in the code is equal to the number of irreducible polynomials in the factorization of the p -polynomial.
- (2) The number of members in each cycle is equal to the degree of the corresponding irreducible polynomial in the factorization.
- (3) The weights of the representative roots of each factor in its NB cartesian representation determine the weight distribution of the code.

BCH and Golay codes (which are essentially cyclic codes) and their weight distributions are studied with the help of their NB p -polynomials. Finally, self dual cyclic codes are shown to have NB p -polynomial representations which split in $GF(2^k)$, with the result that the number of cycles in a self dual code is equal to the number of Frobenius classes in $GF(2^k)$, with the number of members in each cycle being equal to the order of the Frobenius class.

7.2 Suggestions for Further Work

In this section, we give some suggestions for further studies on GSFs, their polynomial representations, and possible applications.

- (1) In this thesis we have characterized the standard classes of LBFs and β -SD/ASD BFs in terms of their algebraic structures and transform coefficients. But there are other classes of BFs, such as linearly separable functions, bent functions etc., on which a study may be conducted in terms of their GP coefficients to see whether they possess any algebraic structure or whether their GP coefficients satisfy certain constraints which help in their identification in the transform domain. Further, in this thesis, characterization of β -SD/ASD BFs of only upto 4 variables is considered. However, it is possible to derive constraints for higher variable cases also, by proceeding on similar lines as in the cases considered here. This work may be pursued.
- (2) Implementational details of finite field models for synthesizing BFs may also be worked out.
- (3) Various decoding procedures based on the standard array have been proposed in this thesis. It would be interesting to take up a structural study of GPs which represent these decoding procedures.
- (4) Study of GSF representations of codes has been restricted to the class of linear block codes. It is possible to have GSF representations of codes which are not linear. Extension to codes with memory, especially convolutional codes, is also possible. Research in these directions may be initiated.
- (5) Multi-dimensional GSFs have been studied with reference to 2-D GSFs only. Study of GSFs of higher dimensions and their applications may be undertaken.
- (6) Applications of GSF theory in this thesis has been restricted to the areas of switching functions and error control codes. However, they can be utilized in other areas too. Applications of GSFs in permutation and substitution networks may be

investigated for their use in cryptography. Representation and processing of pictorial information in the form of GPs is another potential application of GSFs. Study of both 1-D and 2-D GSF representations of pictorial data may be helpful in the compact representation of images in the form of structured polynomials. Applications of GSFs in fault tolerant computing systems may also be investigated.

- (7) Study of SPs has been mainly centered around quasi-cyclic and cyclic codes. SPs of other well known codes may also be taken up for study to see whether they help in the characterization of these codes.
- (8) Study of SPs in this thesis has been confined to the field of coding. Possibility of their utility in other areas may be explored. For example, it is known that the class of ASD BF's of a given number of variables has the structure of a vector space. This allows all ASD BF's of a given number of variables to be considered as the roots of a single LP, thus giving rise to a compact representation of these functions. Structure of such polynomials may be investigated.
- (9) Throughout this thesis, we have worked with GSFs and their GP representations over finite fields of characteristic 2. It will be worthwhile to conduct a study of GSFs mapping from $GF(p^k)$ to $GF(p^n)$, where p is any prime. This may lead to significant generalizations of results obtained in this thesis for $p = 2$.

APPENDIX A

MATHEMATICAL BACKGROUND

This appendix gives a brief introduction to various mathematical topics relevant to this thesis. The material presented here is available in any of the standard text books on Algebra, Finite Fields and Coding Theory and is compiled for easy reference. This appendix is divided into three sections. First section reviews the *basic algebraic structures* which are employed in the thesis. Second section is on *discrete Fourier transform over finite fields* which includes a discussion on conjugacy constraints. The last section deals with the special class of polynomials known as *linearized polynomials* whose theory is frequently applied in the thesis.

A.1 Basic Algebraic Structures

This section discusses the various algebraic structures which are employed in the thesis. Standard text books on Algebra and Finite Fields such as [34, 35, 36, 27] have been referred to for describing the concepts involved.

A.1.1 Semigroups

The simplest algebraic structure that one can think of is a set D with a binary operation (η) defined on any two elements in the set, satisfying the following axioms:

1. The set D is closed under the operation η .
2. The operation η is associative. ie., if a , b and c are three elements in the set D , then $\eta(a, \eta(b, c)) = \eta(\eta(a, b), c)$.

A structure satisfying the above is called a *semigroup*.

[The operation η is said to be *commutative* or *Abelian* if $\eta(a,b) = \eta(b,a)$.]

Example A.1.1: For the illustration of a semigroup, consider an electrical signal sampled at regular intervals. Let the sampled values be denoted by f_1, f_2, f_3, \dots . The index set of f is the set of all positive integers. We define addition as the binary operation (η) on this set. It can be seen that this operation is associative. Thus this set is an example of a semigroup.

A.1.2 Monoids

An *identity* element is defined as some element 'e' in the set such that $\eta(e,a) = a$ for any element 'a' in the set.

If the identity element is also added to the semigroup structure, we get another algebraic structure called a *monoid*.

Example A.1.2: In Example A.1.1, if we also include f_0 to the sampled values, then the index set of f becomes the set of all natural numbers 0, 1, 2, This has the structure of a monoid in which the element '0' is the identity.

A.1.3 Groups

Inverse of an element a is defined as a^{-1} such that $\eta(a, a^{-1}) = e$ where e is the identity element.

If, for every element a , there exists an inverse a^{-1} in the monoid, then it has the structure of a *group*.

Example A.1.3: If we add to the sampled signal mentioned in Example A.1.2, sampled values with negative indices also, then the index set becomes the set of integers $\{\dots -2, -1, 0, 1, 2, \dots\}$ which has the structure of a group.

The *order* of a group is the number of elements in the group.

A.1.3.1 Cyclic Groups

A multiplicative group is said to be *cyclic* if there is an element a in the group such that for any b in the group, there is some integer j with $b = a^j$. Such an element a is called a *generator* of the cyclic group.

A.1.3.2 Subgroups

A subset of a group which itself has a group structure with respect to the operations of the group is known as a *subgroup*.

Subgroups other than the trivial subgroups [the identity element and the whole group itself are examples of trivial subgroups] are known as *nontrivial subgroups*.

The order of a finite subgroup divides the order of the finite group which contains it.

It may be noted that, depending on whether the set is finite or infinite and whether the binary operation is commutative or non-commutative, additional adjectives can be added to the basic structural names of the algebraic structures being discussed in this appendix.

A.1.3.3 Cosets

Let H be a subgroup of a group G and let ' a ' be in G . Then the set $Ha = \{ba \mid b \in H\}$ is called a *right coset* of H . Similarly the set $aH = \{ab \mid b \in H\}$ is called a *left coset*. For commutative groups, left and right cosets are identical and we will simply call them as *cosets*. The number of elements in a coset is same as the order $|H|$ of the subgroup H . Two cosets of H in G are either disjoint or identical. The number of cosets of H in G is called the *index* of H in G and is denoted as $[G:H]$. The index of any subgroup divides the order $|G|$ of the group.

A.1.4 Morphisms

Morphisms are mappings that preserve the operations between algebraic structures.

A mapping f of an algebraic structure such as a group G_1 into another group G_2 , is called a *Homomorphism* of G_1 into G_2 if it preserves the operation of G_1 , i.e., if we denote '*' and '.' as the operations of G_1 and G_2 respectively, then f preserves the operation of G_1 if for all $a, b \in G_1$, we have $f(a * b) = f(a).f(b)$. If f is onto, then f is called an *Epimorphism*. If f is a one-to-one homomorphism of G_1 onto G_2 , then f is called an *Isomorphism*. An isomorphism of G_1 onto G_1 is called an *Automorphism*.

A.1.5 Rings

So far we discussed sets with one operation. Now we consider sets with two operations. Let us consider one such structure R with the operations of addition (denoted by '+') and multiplication (denoted by '.') satisfying the following axioms:

- R1 $(R, +)$ is an Abelian group.
- R2 $(R, .)$ is a semigroup.
- R3 Multiplication distributes over addition.

i.e., if a, b and $c \in R$, then

$$a.(b+c) = a.b + a.c$$

$$\text{and} \quad (a+b).c = a.c + b.c.$$

The above structure is called a *ring*. If $a.b = b.a$, then R is called a *commutative/Abelian ring*.

Example A.1.4: The set of all even integers with operations '+' and '.' is an example of an infinite commutative ring.

If $(R, .)$ is a monoid, then it is called a *Ring with identity*.

Example A.1.5: The set of all integers with operations '+' and '.' is an example of an

infinite commutative ring with identity.

Example A.1.6: The set of all real matrices with matrix addition and matrix multiplication as the two operations is an example of an infinite non-commutative ring with identity (non-commutative, since matrix multiplication is non-commutative).

Example A.1.7: The set of integers modulo 4, ie., $\{0, 1, 2, 3\}$ is an example of a finite commutative ring with identity. We see that the order of this ring is 4 and is finite.

An important property of a ring is the presence of *zero divisors* in it. Nonzero elements in the ring, which when multiplied gives zero, are called *zero divisors*. The elements in the ring can thus be broadly classified into *zero divisors* and *nonzero divisors*. The nonzero divisors are called *regular elements*. There is a subset of the regular elements called *unit elements*. These are elements of the ring possessing multiplicative inverse.

Example A.1.8: In the ring of integers modulo 4, considered in Example A.1.7, elements 1 and 3 are unit elements (since they possess multiplicative inverses which are 1 and 3 respectively). 2 is a nontrivial/proper zero divisor, since $2 \cdot 2 = 0$ modulo 4 (0 being the trivial zero divisor).

It may be noted that the set of unit elements with multiplication operation possesses a group structure.

A.1.5.1 Subrings

A subset S of a ring R is called a *subring* of R , if S is closed under '+' and '.', and is itself having the structure of a ring under these operations.

A.1.5.2 Ideals

A subset J of a commutative ring R is called a *two-sided ideal* or simply an *ideal*, if

- (i) J is a subring of R , and
- (ii) for all $a \in J$, $r \in R$, we have (a) $ar \in J$ and (b) $ra \in J$.

In non-commutative rings, depending on whether condition (ii, a) is satisfied or whether (ii, b) is satisfied, we have a *right ideal* and a *left ideal* respectively.

Principal Ideal

An ideal generated by a single element r of R is called a *Principal ideal*, denoted by $\langle r \rangle$. If every ideal of R is principal, then R is called a *Principal ideal Ring*.

Proper Ideal

The ideal $\langle 0 \rangle$ generated by '0' is $\{0\}$, whereas the ideal $\langle 1 \rangle$ generated by '1' [the multiplicative identity element] is the ring R itself. These are trivial ideals. An ideal which is neither $\langle 0 \rangle$ nor $\langle 1 \rangle$ is called a *proper ideal* in R .

Idempotent

An element $e \neq 1$ in the ring R such that $e^2 = e$ is called an *idempotent* element in R . If two elements e_i and e_j are such that $e_i e_j = 0$, then they are said to be *orthogonal*. Orthogonal idempotents of R generate proper ideals. We call these ideals as *orthogonal ideals*.

Maximal Ideal

An ideal J_m in R is called a *maximal ideal* if $J_m \neq \langle 1 \rangle$ and there is no ideal J such that $J_m \subset J \subset \langle 1 \rangle$.

Minimal Ideal

An ideal which does not contain any smaller nonzero ideal is called a *minimal ideal*.

A.1.6 Integral Domains

A commutative ring with identity and without proper zero divisors is called an *integral domain*.

Example A.1.9: In the set of integers, with '+' and '.' as the two operations, we see that the only zero divisor is the trivial 0. Thus there are no proper zero divisors in this set and hence is an example of an integral domain.

A.1.7 Fields

A ring in which all the nonzero elements possess a multiplicative inverse is called a *field*.

Example A.1.10: An example of an infinite field is the set of all real numbers with the operations of addition and multiplication.

Example A.1.11: An example of a finite field is the set of integers modulo a prime number 'p' with the operations of addition and multiplication modulo p.

It may be noted that the order of any finite field is some power of a prime number 'p'. [Finite fields are named as *Galois Fields* after their discoverer Evariste Galois. We use the terms '*finite field*' and '*Galois field*' interchangeably and denote them as $GF(\cdot)$.]

A.1.7.1 Subfields

Let F be a field. A subset K of F that is itself a field under the operations of F is called a *subfield* of F . If $K \neq F$, then K is said to be a *proper subfield* of F . A field containing no proper subfields is called a *prime field*. Any finite field of prime order is called a prime field.

A.1.8 Polynomials

A *polynomial* is an expression of the form

$$f(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \quad (\text{A.1.1})$$

where n is a nonnegative integer, a_i 's, $i = 0, 1, \dots, n$, are called the *coefficients* which can be real or complex numbers, and x is a variable.

If the coefficients a_i 's belong to a ring R , then the above expression is called a *Polynomial over a Ring*. We can use f as a designation for the polynomial $f(x)$.

It may be seen that the set of polynomials over a Ring R forms a ring called the *Polynomial Ring over R* , denoted as $R[x]$, under the operations of polynomial addition and polynomial multiplication. Let $f(x)$ and $g(x)$ be two polynomials over a ring R , given by

$$f(x) = \sum_{i=0}^n a_i x^i$$

and

$$g(x) = \sum_{j=0}^m b_j x^j$$

Then (i) *Polynomial addition* of $f(x)$ and $g(x)$ implies

$$f(x) + g(x) = \sum_{i=0}^n (a_i + b_i) x^i, \quad (\text{A.1.2})$$

and

(ii) *Polynomial multiplication* of $f(x)$ and $g(x)$ implies

$$f(x).g(x) = \sum_{k=0}^{n+m} c_k x^k, \quad (\text{A.1.3})$$

where $c_k = \sum_{i+j=k}^n a_i b_j$, $0 \leq i \leq n$, $0 \leq j \leq m$.

The zero element of $R[x]$ is called the *zero polynomial* (denoted as 0) which has all its coefficients zero.

Let $f(x) = \sum_{i=0}^n a_i x^i$ be a polynomial over R not equal to the zero polynomial. Then the leading coefficient $a_n \neq 0$, is called the *leading coefficient of $f(x)$* , a_0 is called the *constant term*, and n is the degree of $f(x)$ denoted as $\deg(f)$. We set $\deg(0)$ to be equal to $-\infty$. Polynomials of degree ≤ 0 are called *Constant Polynomials*. If the ring R has identity element equal to unity, and if the leading coefficient c_n of $f(x)$ is also equal to unity, then such a polynomial $f(x)$ is called a *Monic Polynomial*.

Similar to the definition of $R[x]$, we can also define *Polynomial ring over a Field F* (not necessarily finite), and denote it as $F[x]$. $F[x]$ is an integral domain. This is because $F[x]$ is an integral domain iff F is an integral domain, and every field is an integral domain.

A.1.8.1 Irreducible Polynomials

A polynomial $f \in F[x]$ is said to be *irreducible* over F (or, irreducible in $F[x]$), if f has positive degree, and $f = p.q$ with $p, q \in F[x]$, implies that either p or q is a constant polynomial. In other words, an irreducible polynomial is a polynomial over F of positive degree which allows only trivial factorizations. A polynomial in $F[x]$ of positive degree that is not irreducible over F is said to be *reducible over F* .

Since we will be dealing only with polynomials over fields which are finite, henceforth $F[x]$ would mean a polynomial ring over a *finite field F* .

Let $f \in F[x]$ be a nonzero polynomial and let $f(0) \neq 0$. Then the least positive integer e for which $f(x)$ divides $x^e - 1$, is called the *order* of f , and denoted as $\text{ord}(f)$.

A.1.8.2 Primitive Polynomials

A polynomial f of degree n over a finite field F of characteristic p , is said to be *primitive* iff f is monic, $f(0) \neq 0$, and order of f is equal to $p^m - 1$.

It may be noted that all primitive polynomials are irreducible, but the converse is not true.

Example A.1.12: Let $f = x^4 + x + 1$ which is known to be irreducible over $GF(2)$. The order of f is $15 = 2^4 - 1$, and hence is primitive. On the other hand, let us take $f = x^4 + x^3 + x^2 + x + 1$, which is again an irreducible polynomial over $GF(2)$. The order of f is 5 which is less than $2^4 - 1 = 15$, and hence is not primitive.

Table A.1: List of Primitive Polynomials over $GF(2)$ of degree n ; $2 \leq n \leq 15$

n	Primitive polynomial over $GF(2)$ of degree n
2	2 1
3	3 2
4	4 1
5	5 2
6	6 1
7	7 3
8	8 4 3 2
9	9 4
10	10 3
11	11 2
12	12 6 4 1
13	13 4 3 1
14	14 10 6 1
15	15 1

A list of primitive polynomials over $GF(2)$ of degree n is given in Table A.1, for $2 \leq n \leq 15$. We list only the exponents of x in the table. The constant term '1' is present in all the polynomials and hence is not listed. Thus an entry 7 3 corresponds to the primitive polynomial $x^7 + x^3 + 1$.

Irreducible polynomials are essential for constructing extension fields, as we will see shortly.

A.1.8.3 Roots of Polynomials

An element $\alpha \in F$ is called a *root* or a *zero* of the polynomial $f \in F[x]$, if $f(\alpha) = 0$. Further, an element $\alpha \in F$ is a *root* of the polynomial $f \in F[x]$, iff $(x - \alpha)$ divides $f(x)$.

Let $\alpha \in F$ be a root of the polynomial $f \in F[x]$. If k is a positive integer such that $f(x)$ is divisible by $(x - \alpha)^k$, but not by $(x - \alpha)^{k+1}$, then k is called the multiplicity of α . If $k = 1$, then α is called a *simple root* or a *simple zero* of f , and if $k \geq 2$, then α is called a *multiple root* or *multiple zero* of f .

A.1.8.4 Derivative of Polynomials

If $f(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n \in F[x]$, then the *derivative* f' of f is defined as

$$f' = f'(x) = a_1 + 2 a_2 x + \dots + n a_n x^{n-1} \in F[x]. \quad (\text{A.1.4})$$

The element $\alpha \in F$ is a multiple root of $f \in F[x]$ iff it is a root of both f and f' .

Thus if f' is a constant, then the roots of f are simple.

A.1.9 Extension Fields

Now we come to the topic of finite field extensions. For this we draw an analogy from the real field. We have seen that irreducible polynomials are polynomials which cannot be factorized into factors with coefficients from the base field. For example, $x^2 + 1$

is an irreducible polynomial over the real field. Let us introduce a ' j ' such that $j^2 + 1 = 0$, or $j = \sqrt{-1}$, which does not belong to the real field. Now using this ' j ', we can factorize $x^2 + 1$ into linear factors $(x + j)(x - j)$. By introducing j , we can thus extend the real field into the field of complex numbers. In the larger field of complex numbers, instead of a single number, we have ordered pairs (a, b) , (c, d) etc., corresponding to the complex numbers $(a + jb)$, $(c + jd)$ etc., respectively. Finite fields can also be extended in a similar way.

Example A.1.13: Consider the finite field of order 2, namely $GF(2)$, consisting of $(0, 1)$ and with operations '+' and '.' modulo 2. Now we consider extending $GF(2)$. As in the case of the real field, here also we need an irreducible polynomial. Consider polynomials with coefficients from $GF(2)$, i.e., polynomials over $GF(2)$. Degree 1 polynomials are x and $x + 1$. Degree 2 polynomials are x^2 , $x^2 + 1$, $x^2 + x + 1$ and $x^2 + x$. It can be readily seen that, of these, $x^2 + x + 1$ is an irreducible polynomial over $GF(2)$. So we use this polynomial for extending $GF(2)$. Let us introduce an a such that $a^2 + a + 1 = 0$. We can thus extend the base/ground field of $\{0, 1\}$ to $\{0, 1, a, a+1\}$ consisting of 4 elements. This extension field is denoted as $GF(2^2)$.

In general, using an irreducible polynomial of degree n , we can extend the base field $GF(p)$ to a field of order p^n and the extension field is denoted as $GF(p^n)$ where p is a prime.

A.1.9.1 Representation of Elements in an Extension Field

We have various ways of representing the elements of an extension field. One is the *polynomial notation*, as a polynomial in x . Thus the four elements of $GF(2^2)$ in polynomial notation are $0 + 0x$, $0x + 1$, $1x + 0$ and $1x + 1$. Secondly, if we represent them using only the coefficients of x , we have an *ordered tuple notation* or *cartesian representation*. In this form, the above becomes $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$. A third notation is the *polar*

representation where the elements are denoted by the power of an element a in the extension field. We can see that all the nonzero elements of $GF(p^n)$ can be denoted as $a^0, a, a^2, \dots, a^{p^n-2}$, i.e., they can be generated by a power of a . Such an element a in the field whose powers generate all the nonzero elements of the field is called a *primitive element*. For the sake of uniformity, we can denote the element 0 also as a power of a , namely $a^{-\infty}$.

Example A.1.14: The elements of $GF(2^2)$ represented in polar form are $a^{-\infty}, a^0, a$ and a^2 whereas the same represented in cartesian form are 00, 01, 10 and 11 respectively.

The prime number 'p' is called the *characteristic* of the field. Thus in Example A.1.14, $p = 2$.

A.1.10 Frobenius Cycles

If we consider the roots of an irreducible polynomial over the real field, we see that they occur in complex conjugate pairs, i.e., if $(a + jb)$ is a root, then $(a - jb)$ is also a root. This property can be applied to irreducible polynomials over finite fields also. Thus if a is a root of an irreducible polynomial over $GF(p)$, then $a^p, a^{p^2}, \dots, a^{p^{i-1}}$ are also roots of the same irreducible polynomial where $a^{p^i} = a$, the powers taken modulo $p^n - 1$ and $a^{p^j} \in GF(p^n)$, $j = 0, 1, \dots, i-1$. The irreducible polynomial whose roots are a^{p^j} , $j = 0, 1, \dots, i-1$, is called the *minimal polynomial* of a^{p^j} . We can denote the set $\{a, a^p, a^{p^2}, \dots, a^{p^{i-1}}\}$ as a *set of conjugate elements* or a *Frobenius cycle*. We can drop a and group the exponents alone in which case the set can be called as a *conjugacy class*.

Example A.1.15: If we take an irreducible polynomial over $GF(2)$ of degree 3, say $x^3 + x + 1$, and if a is a root of this polynomial, then a^2 and a^4 are also roots of the same

polynomial where $a^2 = a^8 = a$ and all $a^{2^j} \in \text{GF}(2^3)$, $j = 0, 1, 2$. Then the set $\{a, a^2, a^4\}$ is a set of conjugate elements or a Frobenius cycle. $\{1, 2, 4\}$ is a conjugacy class.

A.1.10.1 Frobenius Sum

A feature worth noting is the fact that the sum of the elements in a Frobenius cycle belongs to the ground field. This sum is called a *Frobenius sum* and denoted as $\text{frs}(\cdot)$. Thus Frobenius sum of a , where $a \in \text{GF}(p^n)$, is given by

$$\text{frs}(a) = a + a^p + a^{p^2} + \dots + a^{p^{n-1}}, \quad (\text{A.1.5})$$

where $a^{p^i} = a$.

A.1.10.2 Trace

We also define *trace* of an element a , to distinguish it from Frobenius sum, as

$$\text{tr}(a) = a + a^p + a^{p^2} + \dots + a^{p^{n-1}}, \quad (\text{A.1.6})$$

where n is the order of extension of the field to which a^{p^j} 's belong to. If the number of elements in the Frobenius cycle is equal to n , then the trace function and Frobenius sum are identical, otherwise not. However both belong to the ground field.

Example A.1.16: Let us compute the Frobenius sum and trace of a^5 in $\text{GF}(2^4)$ where a is a root of the primitive polynomial $x^4 + x + 1$. The Frobenius cycle containing a^5 is given by $\{a^5, a^{10}\}$.

$$\text{frs}(a^5) = a^5 + a^{10} = 1,$$

whereas

$$\text{tr}(a^5) = a^5 + a^{10} + a^5 + a^{10} = 0.$$

Frobenius sum and trace of an element of an extension field can also be defined with respect to a subfield of it other than the ground field $\text{GF}(p)$. Let $\Theta \in \text{GF}(p^L)$, which is an

extension field of $GF(p)$, and let $GF(Q)$ [where Q is some prime power, say p^n] be a subfield of $GF(p^L)$. From finite field theory, since $GF(p^n)$ is a subfield of $GF(p^L)$, $n|L$.

The set $\{\theta, \theta^Q, \theta^{Q^2}, \dots, \theta^{Q^{i-1}}\}$, where $\theta^{Q^i} = \theta$, is a Frobenius cycle.

We define

$$\text{frs}(\theta) = \theta + \theta^Q + \theta^{Q^2} + \dots + \theta^{Q^{i-1}}, \quad (\text{A.1.7})$$

where $\theta^{Q^i} = \theta$ and

$$\text{tr}(\theta) = \theta + \theta^Q + \theta^{Q^2} + \dots + \theta^{Q^{(L/n)-1}}. \quad (\text{A.1.8})$$

A.1.11 Vector Spaces

Now let us introduce the notion of a vector space. A *vector space* V is defined with respect to a field F . There is a binary operation called *vector addition* (+) defined on the set V and a unary operation called *scalar multiplication* (\otimes) involving multiplication of a vector belonging to V by a scalar belonging to F .

The following axioms are satisfied:

- (1) $(V, +)$ is an Abelian group.
- (2) Let α and $\beta \in F$ and $v \in V$, then
 - (i) There is an identity element '1' of the field F such that $1 \otimes v = v$.
 - (ii) Associative law is satisfied;
ie., $\alpha \otimes (\beta \otimes v) = (\alpha \otimes \beta) \otimes v$.
- (3) Distributive law is satisfied:
ie., if v_1 and $v_2 \in V$, and α and $\beta \in F$, then
 - (i) $\alpha \otimes (v_1 + v_2) = \alpha \otimes v_1 + \alpha \otimes v_2$,
where '+' is the vector addition, and
 - (ii) $(\alpha <+> \beta) \otimes v_1 = \alpha \otimes v_1 <+> \beta \otimes v_1$,
where '<+>' is the field addition.

Example A.1.17: Example of a vector space is the finite field $GF(p^n)$ with vector addition and scalar multiplication with respect to $GF(p)$ as the two operations.

A.1.11.1 Subspaces

A nonempty subset of a vector space V , is a *subspace* if it itself is a vector space under the same operations of vector addition and scalar multiplication with respect to the same field as defined for V . The dimension of a subspace of an n -dimensional vector space is $\leq n$.

Example A.1.18: Two trivial subspaces are the set containing the vector $\{0\}$ and the whole space V . An example of a nontrivial subspace is a linear (n,k) block code which is a k -dimensional subspace of the vector space of n -tuples, where n is the block length, k is the dimension of the code and $k < n$.

A.1.11.2 Notion of Linear Independence

A set of vectors v_1, v_2, \dots, v_n is said to be *linearly dependent*, if there exist elements a_1, a_2, \dots, a_n in the field F , not all zero, such that $a_1 v_1 + a_2 v_2 + \dots + a_n v_n = 0$. They are said to be *linearly independent* if not linearly dependent. No vector in a linearly independent set can be expressed as a linear combination of any other vectors in the set.

A.1.11.3 Basis

A minimal set of linearly independent vectors which generates the vector space is called a *basis*. *Dimension* of the vector space is the number of elements in the basis. All the elements in the vector space can thus be expressed as a linear combination of ' n ' linearly independent vectors. However, there can be more than one linearly independent set, i.e., the basis of a vector space is not unique.

A.1.12 Different Bases for Finite Fields

A finite field has a vector space structure and therefore can be generated by a basis. The number of possible bases for a finite field can be very large. However, we will discuss only two important bases in this subsection, namely, the standard basis and the normal basis.

A.1.12.1 Standard / Polynomial Basis

In constructing $GF(p^n)$ from a primitive irreducible polynomial $p(x)$, we used the basis, $\alpha^0, \alpha, \alpha^2, \dots, \alpha^{n-1}$, where α is a root of $p(x)$. This basis is known as the *standard basis* (SB) or *polynomial basis*.

A.1.12.2 Normal Basis

A normal basis (NB) of $GF(p^n)$ is a set of linearly independent vectors of the form $\gamma, \gamma^p, \gamma^{p^2}, \dots, \gamma^{p^{n-1}}$ consisting of an element $\gamma \in GF(p^n)$ and its conjugates. It has been proved that there exists a NB for every finite field. Since this basis has been employed in this thesis more often, we elaborate on it below:

(A) Advantages of Normal Basis

There are certain implementational advantages of working with NB in finite field squarers and multipliers.

(a) Squaring

Let us consider the squaring of any finite field element expressed in NB. We will limit our discussions to finite fields of characteristic 2. Let 'a' be any element of $GF(2^n)$. Let $\gamma, \gamma^2, \gamma^{2^2}, \dots, \gamma^{2^{n-1}}$ be any set of linearly independent vectors belonging to $GF(2^n)$

elements, is

$$a = a_{n-1}\gamma^{2^{n-1}} + a_{n-2}\gamma^{2^{n-2}} + \dots + a_1\gamma^2 + a_0\gamma, \quad (\text{A.1.9})$$

where a_i 's ($i = 0, 1, \dots, n-1$) $\in \text{GF}(2)$.

With the following facts in mind, that

$$(1) \quad \text{for any } a, b \in \text{GF}(2^n), (a + b)^2 = a^2 + b^2 \quad (\text{A.1.10})$$

(ie., squaring is a linear operation in $\text{GF}(2)$), and

$$(2) \quad \text{for any } a \in \text{GF}(2^n), a^{2^n} = a, \quad (\text{A.1.11})$$

let us square a . Thus

$$\begin{aligned} a^2 &= a_{n-1}\gamma^{2^n} + a_{n-2}\gamma^{2^{n-1}} + \dots + a_1\gamma^{2^2} + a_0\gamma^2 \\ &= a_{n-1}\gamma + a_{n-2}\gamma^{2^{n-1}} + \dots + a_1\gamma^{2^2} + a_0\gamma^2 \\ &= a_{n-2}\gamma^{2^{n-1}} + a_{n-3}\gamma^{2^{n-2}} + \dots + a_1\gamma^{2^2} + a_0\gamma^2 + a_{n-1}\gamma. \end{aligned} \quad (\text{A.1.12})$$

Now omitting the basis elements, and expressing a and a^2 only using a_i 's (the cartesian representation), we have

$$a = a_{n-1} \ a_{n-2} \ \dots \ a_1 \ a_0$$

and

$$a^2 = a_{n-2} \ a_{n-3} \ \dots \ a_1 \ a_0 \ a_{n-1}.$$

This evidently suggests a simple method for squaring a finite field element expressed in NB, namely, by *merely cyclic shifting* the cartesian representation of the element in NB towards the left by one bit.

(b) Multiplication

Now let us consider multiplication of two elements a and b , expressed in NB. Let us denote the product of a and b as c .

$$\text{Let} \quad a = a_{n-1} \ a_{n-2} \ \dots \ a_1 \ a_0$$

$$\text{and} \quad b = b_{n-1} \ b_{n-2} \ \dots \ b_1 \ b_0$$

Now $c = a.b$

$$\begin{aligned} &= c_{n-1} c_{n-2} \dots c_1 c_0 \\ &= [a_{n-1} a_{n-2} \dots a_1 a_0] [b_{n-1} b_{n-2} \dots b_1 b_0], \end{aligned} \quad (A.1.13)$$

where the a_i 's, b_i 's and c_i 's $\in GF(2)$ and c_i 's are some function of a_i 's and b_i 's. ($i = 0, 1, \dots, n-1$),

$$\text{ie.,} \quad c_i = f(a_i, b_i). \quad (A.1.14)$$

Let the coefficient c_{n-1} corresponding to the highest power of γ be expressed as

$$c_{n-1} = f(a_{n-1} a_{n-2} \dots a_1 a_0; b_{n-1} b_{n-2} \dots b_1 b_0)$$

Now let us square c

$$c^2 = a^2 b^2$$

$$\begin{aligned} \text{ie.,} \quad c^2 &= [c_{n-2} c_{n-3} \dots c_1 c_0 c_{n-1}] \\ &= [a_{n-2} a_{n-3} \dots a_1 a_0 a_{n-1}] [b_{n-2} b_{n-3} \dots b_1 b_0 b_{n-1}], \end{aligned} \quad (A.1.15)$$

since squaring of an element as mentioned earlier, is a cyclic shift of the corresponding cartesian representations towards left.

Now, the coefficient corresponding to the highest power of γ of c^2 is c_{n-2} , which can be obtained by the same function $f(\cdot)$ as used to calculate c_{n-1} , but with the a_i 's and b_i 's which are input to the function, cyclically shifted towards left by one bit. In this manner, all the c_i 's, $i = 0, 1, \dots, n-1$, can be calculated.

Implementation Schemes

The implementation of the multiplier can be done in two ways:

(a) Serial Implementation

In this implementation, store the numbers to be multiplied, namely, a and b , in two registers A and B , the output of which is input to a block F which realizes the function $f(\cdot)$, to get c_{n-1} at the first clock pulse. At the arrival of the next clock pulse, cyclically shift left the contents of A and B by one bit to get c_{n-2} at the output of F . This process is

repeated till we get all the c_i 's, $i = 0, 1, \dots, n-1$. Thus the serial implementation requires n clock periods and one functional block to perform one multiplication operation in NB.

(b) Parallel Implementation

We can have a parallel implementation hardware, in which case, the speed is increased by n times, and we can perform multiplication in one clock period. In this implementation, we have n identical blocks, each block having the same components as in (a). The input to the first block is the a_i 's and b_i 's which realize c_{n-1} . The remaining $n-1$ blocks realize $c_{n-2}, c_{n-3}, \dots, c_0$ respectively. The input to these blocks will be cyclic shifted versions of a_i 's and b_i 's, each succeeding input differing from the preceding input by one cycle. Thus in this case, the speed is increased by a factor of n , but the hardware requirement also has increased by the same factor.

(B) Conversion from Standard Basis to Normal Basis

It is frequently required to convert elements represented in standard basis (SB) to NB and vice versa, to do operations like squaring. So it is desirable to have an idea as to how these conversions are carried out. The procedure is as follows:

First, one has to find a suitable set of linearly independent vectors of the form $\gamma, \gamma^2, \gamma^{2^2}, \dots, \gamma^{2^{n-1}} \in GF(2^n)$ which can constitute a NB for $GF(2^n)$. We know that each of these vectors can be represented as a linear combination of the SB vectors, $\alpha^0, \alpha, \alpha^2, \dots, \alpha^{n-1}$, where α is a primitive element of $GF(2^n)$. Thus, this relation is expressed in matrix form, as

$$\begin{bmatrix} \gamma^{2^{n-1}} \\ \gamma^{2^{n-2}} \\ \gamma^{2^{n-3}} \\ \vdots \\ \gamma \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdot & \cdot & a_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \alpha^{n-1} \\ \alpha^{n-2} \\ \cdot \\ \cdot \\ \alpha^0 \end{bmatrix}$$

or,

$$\gamma = \underline{A} \alpha, \quad (\text{A.1.16})$$

where the a_{ij} 's, $1 \leq i, j \leq n$, $\in \text{GF}(2)$.

From this relation, we find the linear transformation matrix which expresses the SB vectors in terms of the NB vectors, by simply inverting the matrix \underline{A} in the right hand side of (A.1.16).

$$\text{Thus } \alpha = \underline{A}^{-1} \gamma, \quad (\text{A.1.17})$$

ie.,

$$\begin{bmatrix} \alpha^{n-1} \\ \alpha^{n-2} \\ \alpha^{n-3} \\ \vdots \\ \alpha^0 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ b_{31} & b_{32} & \cdot & \cdot & b_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nn} \end{bmatrix} \begin{bmatrix} \gamma^{2^{n-1}} \\ \gamma^{2^{n-2}} \\ \cdot \\ \cdot \\ \gamma \end{bmatrix}$$

where the $n \times n$ matrix \underline{A}^{-1} of b_{ij} 's $\in \text{GF}(2)$, is the inverse of \underline{A} .

Now, to find the cartesian representation of an element $d \in \text{GF}(2^n)$, in NB, first find its cartesian representation in SB, ie.,

$$d = d_{n-1}\alpha^{n-1} + d_{n-2}\alpha^{n-2} + \dots + d_1\alpha + d_0\alpha^0,$$

where $d_i \in \text{GF}(2)$.

To express d in NB, it is required only to multiply the $1 \times n$ matrix of d_i 's with the $n \times n$ matrix \underline{A}^{-1} . For finite fields of characteristic 2, this amounts to taking the linear combination of those rows of \underline{A}^{-1} , corresponding to $d_i = 1$.

For conversion of an element represented in NB to SB, we follow the same procedure as above, but with \underline{A}^{-1} replaced by \underline{A} .

Number of Different Normal Bases in a Finite Field

As mentioned earlier, it has been proved that there exists a NB for every finite field. The number of different NBs in a finite field $\text{GF}(p^n)$ over $\text{GF}(p)$ has also been derived elsewhere [27]. This number, say $\#$, is given to be

$$\# = (1/n) \phi_p(f), \quad (\text{A.1.18})$$

where $f = x^n - 1$ and $\phi_p(f)$ is the analog of Euler's phi function, which is the number of polynomials over $\text{GF}(p)$ that are of smaller degree than f , as well as relatively prime to it. Further, if the degree of $f \geq 1$, then

$$\phi_p(f) = p^n \cdot (1 - p^{-n_1})(1 - p^{-n_2}) \dots (1 - p^{-n_r}), \quad (\text{A.1.19})$$

where n_i are the degrees of the distinct monic irreducible polynomials in the canonic factorization of f over $\text{GF}(p)$.

As examples, we find the number of different NBs in $\text{GF}(2^9)$ and $\text{GF}(2^{12})$ over $\text{GF}(2)$.

Example A.1.19: $n = 9$

$$x^9 + 1 = (x + 1)(x^2 + x + 1)(x^6 + x^3 + 1).$$

$$\phi_2(x^9 + 1) = 2^9(1 - 2^{-1})(1 - 2^{-2})(1 - 2^{-6}) = 18.$$

Therefore, the number of different NBs of $\text{GF}(2^9) = \# = 189/9 = 21$.

Example A.1.20: $n = 12$

$$x^{12} + 1 = (x + 1)^4(x^2 + x + 1)^4.$$

$$\phi_2(x^{12} + 1) = 2^{12}(1-2^{-1})(1-2^{-2}) = 1536.$$

Therefore, the number of different NBs of $\text{GF}(2^{12}) =$

$$\# = 1536/12 = 128.$$

The number of different NBs ($\#$) of $\text{GF}(2^n)$ over $\text{GF}(2)$ are tabulated in Table A.2 for $2 \leq n \leq 15$.

Table A.2: Number of Different Normal Bases in $\text{GF}(2^n)$; $2 \leq n \leq 15$

n	#
2	1
3	1
4	2
5	3
6	4
7	7
8	16
9	21
10	48
11	93
12	128
13	315
14	448
15	675

Before concluding this section, we give the definition of an algebra as below:

A.1.13 Algebra

A ring R endowed with a vector space structure over a field F is called an *Algebra* A over F if

$$\alpha \otimes (v_1 \cdot v_2) = (\alpha \otimes v_1) \cdot v_2 = v_1 \cdot (\alpha \otimes v_2)$$

where $\alpha \in F$, $v_1, v_2 \in R$ and ' \cdot ' denotes the multiplication operation in the ring.

In the next section, we will have a brief discussion on *discrete Fourier transform over finite fields* whose theory is frequently applied in this thesis.

A.2 Discrete Fourier Transform (DFT) over Finite Fields

Fourier transforms (FT) are useful in the study of real or complex valued signals when the time variable is continuous. When the time variable is discrete, discrete Fourier transform (DFT) plays the same role as FT. FT and DFT are widely used in signal processing and communication. The concept of DFT can be extended to vectors over finite fields also. To start with, we define DFT over the complex field as follows:

The DFT of a vector of length N over the complex field is defined as

$$C_k = \sum_{i=0}^{N-1} e^{j \frac{2\pi}{N} ik} c_i, \quad (A.2.1)$$

$k = 0, 1, \dots, N-1$, where $j = \sqrt{-1}$. The Fourier kernel $e^{j \frac{2\pi}{N} ik}$ is called an N^{th} root of unity in the field of complex numbers.

In the finite field $GF(q^m)$, an element α of order n is an n^{th} root of unity. We immediately see that elements of all orders do not exist in $GF(q^m)$. Elements of $GF(q^m)$ can only have orders which divide $q^m - 1$. Thus, unlike in the case of DFT over the complex field, DFT over a finite field can be defined only for vectors whose lengths n divide $q^m - 1$. Now, the DFT of a vector over a finite field can be defined as follows:

Let $\underline{v} = [v_0 \ v_1 \ v_2 \ \dots \ v_{n-1}]$ be a vector where $v_i, i = 0, 1, \dots, n-1, \in GF(q)$ and where $n|q^m-1$. Let α be an element of order n in $GF(q^m)$. Then the finite field DFT of the vector \underline{v} , is the vector $\underline{V} = [V_0 \ V_1 \ V_2 \ \dots \ V_{n-1}]$ over $GF(q^m)$, where

$$V_j = \sum_{i=0}^{n-1} \alpha^{ij} v_i, j = 0, 1, \dots, n-1, \quad (A.2.2)$$

where $i \cdot j$ is taken modulo q^m-1 .

Similar to the existence of complex valued FT for real valued functions, there exist $GF(q^m)$ valued DFTs for $GF(q)$ valued signals.

We naturally call the discrete index ' i ' as *time*, and \underline{v} as the *time domain function*, and the discrete index ' j ' as *frequency* and \underline{V} as the *frequency domain function*.

Before concluding this section, we look into the conjugacy relations among DFT coefficients:

A.2.1 Conjugacy Constraints

As mentioned earlier, the DFT of a vector over $GF(q)$ of length n takes values in an extension field $GF(q^m)$. However, the inverse DFT of any arbitrary n length vector over $GF(q^m)$ does not in general give a vector whose components belong to $GF(q)$. Thus the frequency domain vector should satisfy certain constraints in order to ensure that all the components of its inverse DFT (ie., the time domain vector) lie in $GF(q)$. These constraints are similar to the case of Fourier transform over the complex field. Over the complex field, a spectrum $F(f)$ has a real valued inverse Fourier transform iff $F^*(-f) = F(f)$, where '*' denotes complex conjugate. Analogously, we can define constraints for the finite field case also. Thus we state the following theorem known as *conjugacy theorem* without proof (For proof, see [20]):

Theorem A.2.1: Let $\underline{V} = [V_0 \ V_1 \ V_2 \ \dots \ V_{n-1}]$ be a vector of length n over $GF(q^m)$ where $n|q^m-1$. Then the inverse DFT \underline{v} is a vector over $GF(q)$ iff the following equations are

satisfied:

$$(V_j)^q = V_{jq \bmod n}, \quad j = 0, 1, \dots, n-1. \quad (\text{A.2.3})$$

In the next section, the last in this appendix, we discuss the theory of linearized polynomials which finds an important place in this thesis.

A.3 Linearized Polynomials

This special class of polynomials assumes considerable importance in this thesis and hence an elaborate study of its relevant theory is in order. Thus we devote this section to the description of the available theory and results on this topic, which are relevant to this thesis.

Linearized polynomials (LPs) were first investigated by Ore [23, 24, 25, 26] who called them as *p-polynomials*. However this term was later on restricted to those polynomials in this class whose coefficients belonged to the ground field $GF(p)$, p being a prime number, and the general term of '*q-polynomials over $GF(q^m)$* ', where q is a prime power, was used. Berlekamp [28] has called these polynomials as '*linearized polynomials over $GF(q^m)$* ' for reasons which will become evident shortly.

The theory of LPs described in this section is on the lines of Lidl and Niederreiter [27] with occasional references to Berlekamp [28] and MacWilliams and Sloane [19].

A.3.1 Definition and Terminology

Let q denote a power of a prime, say, $q = p^a$.

Then a *q-polynomial over $GF(q^m)$* or alternately, a *LP over $GF(q^m)$* can be defined as a polynomial of the form

$$F(x) = \sum_{i=0}^h a_i x^{q^i}, \quad (\text{A.3.1})$$

where the coefficients $a_i \in GF(q^m)$.

If the coefficients are from $GF(q)$, the corresponding polynomial will be called a q -polynomial over $GF(q)$.

Our studies are limited to $q = p = 2$. Thus in this thesis, the terms p -polynomials over $GF(p^m)$ and LPs are used interchangeably wherever needed. Some authors define the polynomial to be monic, ie., the leading coefficient $a_h = 1$. However we use the terms, p -polynomials over $GF(p^m)$ and LPs, for both monic as well as nonmonic cases. If the coefficients belong to the ground field, the same will be called simply as a p -polynomial or a q -polynomial.

LPs derive its terminology from its linearity property stated in the following theorem:

Theorem A.3.1: Let $GF(q^s)$ be an arbitrary extension of $GF(q^m)$ where $s \geq m$ and q is a prime power. Let β_1 and $\beta_2 \in GF(q^s)$. Then,

$$(1) \quad F(\beta_1 + \beta_2) = F(\beta_1) + F(\beta_2), \quad (A.3.2)$$

$$(2) \quad F(c\beta_1) = c F(\beta_1), \text{ where } c \in GF(q). \quad (A.3.3)$$

Proof: (1)
$$F(\beta_1 + \beta_2) = \sum_{i=0}^h a_i (\beta_1 + \beta_2)^{q^i} = \sum_{i=0}^h a_i (\beta_1^{q^i} + \beta_2^{q^i})$$
$$= F(\beta_1) + F(\beta_2),$$

since $(\beta_1 + \beta_2)^{q^i} = \beta_1^{q^i} + \beta_2^{q^i}$ in fields of characteristic p , for all $\beta_1, \beta_2 \in GF(q^s)$, and q a power of the prime p .

$$(2) \quad F(c\beta_1) = \sum_{i=0}^h a_i (c\beta_1)^{q^i} = c \sum_{i=0}^h a_i \beta_1^{q^i} = c F(\beta_1),$$

since $c^{q^i} = c$ for all $c \in GF(q)$, and $i > 0$.

Q.E.D.

Because of the above property, the LP is said to induce a linear operator on $GF(q^s)$, if

$GF(q^s)$ is considered as a vector space over $GF(q)$.

A.3.2 Roots of a Linearized Polynomial

We now discuss about the roots of a LP. The roots of a LP over $GF(q^m)$ may not lie in the same field. They may belong to an extension of $GF(q^m)$ also. The next result is about the nature of the set of roots of a LP over $GF(q^m)$, which lies in $GF(q^s)$, $s \geq m$.

Theorem A.3.2: Let $F(x)$ be a nonzero LP over $GF(q^m)$ and let all the roots of $F(x)$ lie in the extension field $GF(q^s)$ of $GF(q^m)$ where $s \geq m$. Then this set of roots forms a linear subspace of $GF(q^s)$ where $GF(q^s)$ is considered as a vector space over $GF(q)$. Further, each root of $F(x)$ has the same multiplicity which is 1 or a power of q .

Proof: Let β_1 and β_2 be two roots of $F(x)$ which belong to $GF(q^s)$. Thus

$$F(\beta_1) = F(\beta_2) = 0.$$

Now, from the linearity property of $F(x)$, $F(\beta_1 + \beta_2) = F(\beta_1) + F(\beta_2)$ is also equal to 0, meaning $\beta_1 + \beta_2$ is also a root of $F(x)$. In general, any linear combination of the roots with coefficients from $GF(q)$ is also a root. In other words, the roots form a linear subspace of $GF(q^s)$, where $GF(q^s)$ is considered as a vector space over $GF(q)$.

Now, let the derivative of $F(x)$ be denoted as $F'(x)$. Then it may be seen that $F'(x)$ is equal to a_0 . Thus if $a_0 \neq 0$, then $F(x)$ has only simple roots. Otherwise we have, $a_0 = a_1 = \dots = a_{t-1} = 0$, but $a_t \neq 0$ for some $t \geq 1$. Therefore

$$F(x) = \sum_{i=t}^h a_i x^{qi}$$

Now since $a_i \in GF(q^m)$, it satisfies the relation $a_i q^{mt} = a_i$. Thus $F(x)$ may be written as

$$F(x) = \sum_{i=t}^h a_i q^{mt} x^{qi} = \left(\sum_{i=t}^h a_i q^{(m-1)t} x^{q^{i-t}} \right)^q,$$

which is the q^t th power of a LP having only simple roots. Thus each root of $F(x)$ has the

same multiplicity q^t .

Q.E.D.

The next theorem is a partial converse to the above theorem.

Theorem A.3.3: Let U be an h -dimensional subspace of $GF(q^m)$, considered as a vector space over $GF(q)$. Then for any non-negative integer t , the polynomial

$$F(x) = \prod_{\beta \in U} (x - \beta)^{q^t}$$

is a LP over $GF(q^m)$.

Proof: Since the q^t th power of a LP over $GF(q^m)$ is also a LP, we need to consider only the case $t = 0$.

ie., It is sufficient to prove that $F(x)$ is of the form

$$F(x) = \prod_{\beta \in U} (x - \beta) = x^{q^h} + a_{h-1} x^{q^{h-1}} + \dots + a_0 x, \quad (\text{A.3.4})$$

where h is the dimension of the subspace.

Let $(\beta_1, \beta_2, \dots, \beta_h)$ be a basis of U over $GF(q)$. Since β_i , $i = 1, 2, \dots, h$, is a basis, the matrix

$$\underline{\Delta} = \begin{bmatrix} \beta_0 & \beta_0^q & \beta_0^{q^2} & \cdot & \cdot & \beta_0^{q^{h-1}} \\ \beta_1 & \beta_1^q & \beta_1^{q^2} & \cdot & \cdot & \beta_1^{q^{h-1}} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \beta_{h-1} & \beta_{h-1}^q & \beta_{h-1}^{q^2} & \cdot & \cdot & \beta_{h-1}^{q^{h-1}} \end{bmatrix} \quad (\text{A.3.5})$$

is invertible, since $\det |\Delta|$ is known to be nonzero if β_i , $i = 0, 1, \dots, h-1$, are linearly independent [27]. Thus there is a solution a_0, a_1, \dots, a_{h-1} in $GF(q^m)$ to the equations

$$\begin{aligned} \beta_0^q + a_0 \beta_0 + a_1 \beta_0^q + a_2 \beta_0^{q^2} + \dots + a_{h-1} \beta_0^{q^{h-1}} &= 0 \\ \beta_1^q + a_0 \beta_1 + a_1 \beta_1^q + a_2 \beta_1^{q^2} + \dots + a_{h-1} \beta_1^{q^{h-1}} &= 0 \\ &\vdots \\ \beta_{h-1}^q + a_0 \beta_{h-1} + a_1 \beta_{h-1}^q + a_2 \beta_{h-1}^{q^2} + \dots + a_{h-1} \beta_{h-1}^{q^{h-1}} &= 0 \end{aligned} \quad (A.3.6)$$

since $[a_0 \ a_1 \ \dots \ a_{h-1}]^T = \Delta^{-1} [\beta_0^q \ \beta_1^q \ \dots \ \beta_{h-1}^q]^T$, T denoting transpose.

Therefore $\beta_0, \beta_1, \dots, \beta_{h-1}$ are the roots of the LP

$$F(x) = x^{q^h} + a_{h-1} x^{q^{h-1}} + \dots + a_0 x.$$

Now since β_i , $i = 0, 1, \dots, h-1$, are the roots of a LP, any linear combination of β_i , $i = 0, 1, \dots, h-1$, which belongs to U , is also a root of $F(x)$. Hence

$$F(x) = \prod_{\beta \in U} (x - \beta) \quad \text{Q.E.D.}$$

A.3.3 Symbolic Multiplication

For the study of the algebraic properties of LPs, suitable binary operations should be defined on this set such that the set is closed under these operations. One obvious operation is ordinary addition, since addition of two LPs is again a LP. However, ordinary multiplication of two LPs, in general, need not be a LP. Thus a new operation of composition is introduced on this set. This operation of composition is commonly known as *symbolic multiplication* and is denoted as (x) . Thus if $F_1(x)$ and $F_2(x)$ are two LPs over $GF(q^m)$, then their symbolic product defined as $F_1(x) (x) F_2(x) = F_1(F_2(x))$ is again a LP over $GF(q^m)$. The symbolic multiplication is, in general, not commutative.

A.3.4 Dual Polynomials

For every monic LP over $GF(q^m)$ of degree q^k , which divides $x^{q^m} - x$, there exists a unique monic LP called a *dual polynomial* [28] of degree q^{m-k} , which also divides $x^{q^m} - x$, such that the root space of one polynomial is the range space of the other.

Let $F(x)$ be any monic LP over $GF(q^m)$ of degree q^k , which divides $x^{q^m} - x$. Then all the roots of this polynomial lie in $GF(q^m)$, and they form a subspace of $GF(q^m)$. Let r_1, r_2, \dots, r_k be a basis for the root space of $F(x)$, and let $r_1, r_2, \dots, r_k, r_{k+1}, \dots, r_m$ be a basis for $GF(q^m)$, considered as a vector space over $GF(q)$. Then the dual of $F(x)$, denoted as $F_d(x)$ is defined as a monic LP of degree q^{m-k} , such that $F_d(x)$ is given by

$$F_d(x) = \prod_{c_m=0}^{q-1} \prod_{c_{m-1}=0}^{q-1} \dots \prod_{c_{k+1}=0}^{q-1} \left(x - \sum_{i=k+1}^m c_i F(r_i) \right). \quad (A.3.7)$$

while $F(x)$ is given by

$$F(x) = \prod_{c_k=0}^{q-1} \prod_{c_{k-1}=0}^{q-1} \dots \prod_{c_1=0}^{q-1} \left(x - \sum_{i=1}^k c_i r_i \right). \quad (A.3.8)$$

Theorem A.3.4: $F(x)$ and $F_d(x)$ satisfy the relation

$$F_d(x) (x) F(x) = F(x) (x) F_d(x) = x^{q^m} - x. \quad (A.3.9)$$

Proof: $F_d(F(x)) = \prod_{c_m=0}^{q-1} \prod_{c_{m-1}=0}^{q-1} \dots \prod_{c_{k+1}=0}^{q-1} \left(F(x) - \sum_{i=k+1}^m c_i F(r_i) \right),$

on substituting x by $F(x)$ in the expression for $F_d(x)$.

Each product term on the right hand side of the above expression is an affine polynomial (ie., a LP plus a constant term) $A_t = F(x) - u_t$, $t = 1, 2, \dots, q^{m-k}$, all the roots of which lie in $GF(q^m)$. Thus each A_t is a factor of $x^{q^m} - x$. Therefore the right hand side of the expression gives a factorization of $x^{q^m} - x$ and is equal to $x^{q^m} - x$. Similarly it may be

shown that $F(x) (x) F_d(x) = x^{q^m} - x$.

Q.E.D.

A.3.5 Q - Associates

Let $F(x) = \sum_{i=0}^h a_i x^{q^i}$ be a LP over $GF(q^m)$. Then the polynomial

$$f(x) = \sum_{i=0}^h a_i x^i, \quad (\text{A.3.10})$$

is called the *conventional q-associate* of $F(x)$ and $F(x)$ is called the *linearized q-associate* of $f(x)$.

A.3.6 Q-Polynomials over $GF(q)$

LPs whose coefficients are restricted to $GF(q)$ are called *q-polynomials over $GF(q)$* or simply *q-polynomials*. The results available on this topic are discussed below:

A.3.6.1 Algebraic Structure of Q-Polynomials

The operation of symbolic multiplication is commutative in the case of *q-polynomials over $GF(q)$* , besides being associative. Further it distributes with respect to ordinary addition. It may thus be seen that the set of *q-polynomials* forms an *integral domain* under the operations of *symbolic multiplication* and *ordinary addition*.

A.3.6.2. Relating Symbolic Multiplication of Q-Polynomials to Ordinary Polynomial Arithmetic

The operation of symbolic multiplication of *q-polynomials* can be related to conventional polynomial arithmetic in terms of *q-associates*, as stated in the following theorem:

Theorem A.3.5: Let $F_1(x)$ and $F_2(x)$ be two q -polynomials over $GF(q)$ and let their conventional q -associates be $f_1(x)$ and $f_2(x)$ respectively. Then the conventional q -associate of $F_1(x) (x) F_2(x)$ is equal to $f_1(x).f_2(x)$.

Proof: Let $F_1(x) (x) F_2(x) = F(x)$ and let $f_1(x).f_2(x) = f(x)$.

$$\text{Let } f(x) = \sum_i a_i x^i$$

$$f_1(x) = \sum_j b_j x^j$$

and

$$f_2(x) = \sum_k c_k x^k$$

Then their linearized q -associates are

$$F(x) = \sum_i a_i x^{qi}$$

$$F_1(x) = \sum_j b_j x^{qj}$$

and

$$F_2(x) = \sum_k c_k x^{qk}$$

Now

$$F(x) = F_1(x) (x) F_2(x)$$

$$\text{ie., } \sum_i a_i x^{qi} = \sum_j b_j \left(\sum_k c_k x^{qk} \right)^{qj} = \sum_j b_j \sum_k c_k x^{q(k+j)}, \quad (\text{A.3.11})$$

as $c_k \in GF(q)$.

$$\text{Similarly } f(x) = f_1(x).f_2(x)$$

$$\text{ie., } \sum_i a_i x^i = \sum_j b_j x^j \sum_k c_k x^k \quad (\text{A.3.12})$$

(A.3.11) and (A.3.12) are each true iff

$$a_i = \sum_{j+k=i} b_j c_k \text{ for every } i. \quad (\text{A.3.13})$$

Q.E.D.

A.3.6.3 Symbolic Divisibility

Let $F(x)$, $F_1(x)$ and $F_2(x)$ be q -polynomials over $GF(q)$ where $F(x) = F_1(x) (x) F_2(x)$. Then we say that $F_1(x)$ symbolically divides $F(x)$ or that $F(x)$ is symbolically divisible by $F_1(x)$.

Theorem A.3.6: Let $F_1(x)$ and $F(x)$ be q -polynomials over $GF(q)$ with conventional q -associates $f_1(x)$ and $f(x)$ respectively. Then

- (i) $F_1(x)$ symbolically divides $F(x)$ iff $f_1(x)$ divides $f(x)$.
- (ii) If $F_1(x)$ symbolically divides $F(x)$, then $F_1(x)$ also divides $F(x)$ in the ordinary sense. Conversely, if $F_1(x)$ divides $F(x)$ in the ordinary sense, then $F_1(x)$ divides $F(x)$ symbolically.

Proof: (i) is a consequence of Theorem A.3.5.

- (ii) Since $F_1(x)$ symbolically divides $F(x)$, we can write

$$F(x) = F_1(x) (x) F_2(x) \text{ for some } q\text{-polynomial over } GF(q).$$

$$\text{Let } F_2(x) = \sum_{i=0}^h a_i x^{q^i}.$$

$$\begin{aligned} \text{Then we can write } F(x) &= F_2(x) (x) F_1(x) = F_2(F_1(x)) \\ &= a_0(F_1(x)) + a_1(F_1(x))^q + a_2(F_1(x))^{q^2} + \dots + a_h(F_1(x))^{q^h}, \end{aligned} \quad (\text{A.3.14})$$

from which it is evident that $F_1(x)$ divides $F(x)$ in the ordinary sense as $F_1(x)$ is a common factor.

Conversely, suppose $F_1(x)$ divides $F(x)$ in the ordinary sense. We assume $F_1(x) \neq 0$.

Using the division algorithm, we can write

$$f(x) = k(x) \cdot f_1(x) + r(x), \quad (\text{A.3.15})$$

where $\deg(r(x)) < \deg(f_1(x))$.

Then their corresponding linearized q -associates satisfy the relation

$$F(x) = K(x) (x) F_1(x) + R(x). \quad (\text{A.3.16})$$

Now since $F_1(x)$ divides $F(x)$ in the ordinary sense, it divides $K(x) (x) F_1(x)$ and $R(x)$ also in the ordinary sense. But since $\deg(R(x)) < \deg(F_1(x))$, $R(x)$ must be the zero polynomial. Thus $F(x) = K(x) (x) F_1(x)$ meaning that $F_1(x)$ symbolically divides $F(x)$.

Q.E.D.

A.3.6.4 Relating an Irreducible Polynomial with the Irreducible Factors of its Linearized Q -Associate

The order of an irreducible polynomial $f(x)$, which is the least positive integer e such that $f(x)$ divides $x^e - 1$, is related to the degrees of the irreducible factors of its linearized q -associate. This is brought out in the next theorem:

Theorem A.3.7: Let $f(x)$ be irreducible in $GF(q)[x]$ and let $F(x)$ be its linearized q -associate. Then the degree of every irreducible factor of $F(x)/x$ in $GF(q)[x]$ is equal to the order of $f(x)$.

Proof: Let e be the order of $f(x)$. Let $F_1(x) \in GF(q)[x]$ be an irreducible factor of $F(x)/x$ of degree d . Since $f(x)$ divides $x^e - 1$, its linearized q -associate divides $x^{q^e} - x$. Since $F_1(x)$ is a factor of $F(x)/x$, $F_1(x)$ also divides $x^{q^e} - x$. Therefore d should divide e .

By division algorithm, we can write

$$x^d - 1 = g(x).f(x) + r(x), \quad (A.3.17)$$

where $g(x), r(x) \in GF(q)[x]$, and $\deg(r(x)) < \deg(f(x))$.

Turning to their linearized q -associates, we get

$$x^{q^d} - x = G(x) (x) F(x) + R(x), \quad (A.3.18)$$

where the capital letters denote the respective linearized q -associates.

Since $F_1(x)$ divides $x^{q^d} - x$ and $G(x) (x) F(x)$, it also divides $R(x)$.

If $r(x)$ is not the zero polynomial, then $r(x)$ and $f(x)$ are relatively prime. Therefore there

exist polynomials $k(x)$ and $s(x) \in GF(q)[x]$ such that $r(x).s(x) + f(x).k(x) = 1$.

Turning to their linearized q -associates, we get

$$R(x) (x) S(x) + F(x) (x) K(x) = x. \quad (A.3.19)$$

Since $F_1(x)$ divides $R(x)$ and $F(x)$, it follows that $F_1(x)$ divides x , which is impossible.

Thus $r(x)$ is the zero polynomial. Therefore $F_1(x)$ divides $x^d - 1$, and thus e divides d .

Further $d = e$.

Q.E.D.

A.3.6.5 Duals of Q -Polynomials

The existence of a dual q -polynomial over $GF(q^m)$ for any q -polynomial over $GF(q^m)$ which divides $x^{q^m} - x$, was established in Theorem A.3.4. The duals of q -polynomials over $GF(q)$ may be found with ease from their relation with conventional q -associates. Let $F(x)$ be a q -polynomial over $GF(q)$ which divides $x^{q^m} - x$, and let $F_d(x)$ be its dual. Let $f(x)$ and $f_d(x)$ be their respective conventional q -associates. Then we know that the following operations are equivalent:

$$x^{q^m} - x = F(x) (x) F_d(x). \quad (A.3.20)$$

$$x^m - 1 = f(x).f_d(x). \quad (A.3.21)$$

Thus $F_d(x)$ may be calculated by first finding its conventional q -associate $f_d(x)$ from the relation $(x^m - 1)/f(x)$, and then taking its linearized q -associate as $F_d(x)$.

A.3.6.6 Symbolic Irreducibility and Symbolic Factorizations

A q -polynomial $F(x)$ over $GF(q)$ of degree greater than 1 is said to be *symbolically irreducible* over $GF(q)$ if the only symbolic decompositions of $F(x)$ are of the form $F(x) = F_1(x) (x) F_2(x)$ where $F_1(x)$ and $F_2(x)$ are q -polynomials over $GF(q)$, and one of the factors has degree 1. A symbolically irreducible polynomial is always reducible in the ordinary sense, since x is always a nontrivial factor of any LP of degree greater than 1.

It is evident from the relationship between the q -associates that, a q -polynomial

over $GF(q)$ is symbolically irreducible over $GF(q)$ iff its conventional q -associate $f(x)$ is irreducible over $GF(q)$.

Every q -polynomial over $GF(q)$ of degree greater than 1 has a unique *symbolic factorization* into symbolically irreducible polynomials over $GF(q)$. Further, the symbolic factorization of $F(x)$ can be obtained by factorizing its conventional q -associate $f(x)$ into irreducible polynomials in $GF(q)[x]$, and then taking the linearized q -associates of these irreducible factors to be the symbolically irreducible factors.

We conclude this appendix after a brief description of the special structure of the roots of q -polynomials over $GF(q)$.

A.3.6.7 Structure of the Roots of Q -Polynomials

Let all the roots of a q -polynomial over $GF(q)$ lie in $GF(q^m)$. Then the roots form a linear subspace of $GF(q^m)$, considered as a vector space over $GF(q)$, as per Theorem A.3.2. The roots have the additional property that the q^{th} power of a root is again a root. A subspace M having the property that the q^{th} power of every element of M is again in M , is called a q -modulus, or simply a *modulus*.

Theorem A.3.8: The monic polynomial $F(x)$ is a q -polynomial over $GF(q)$ iff each root of $F(x)$ has the same multiplicity, which is 1 or a power of q , and the roots form a q -modulus.

Proof: The necessity of the conditions follows from Theorem A.3.2. Conversely, the given conditions and Theorem A.3.3 imply that $F(x)$ is a q -polynomial over $GF(q^m)$, an extension field of $GF(q)$.

Now if $U = M$, then we have

$$F(x) = \prod_{\beta \in M} (x - \beta)^{q^t}, \text{ for } t \geq 0. \quad (\text{A.3.22})$$

Now
$$(F(x))^q = \prod_{\beta \in M} (x^q - \beta^q)^{q^t} = \prod_{\beta \in M} (x^q - \beta)^{q^t} = F(x^q), \quad (\text{A.3.23})$$

since if $\beta \in M$, β^q also belongs to M .

If $F(x) = \sum_{i=0}^h a_i x^q{}^i$, then $(F(x))^q = \sum_{i=0}^h a_i^q x^q{}^{i+1}$.

Now we have $((F(x))^q = F(x^q)$,

or
$$\sum_{i=0}^h a_i^q x^q{}^{i+1} = \sum_{i=0}^h a_i x^q{}^{i+1}. \quad (\text{A.3.24})$$

Thus for $0 \leq i \leq h$, we have $a_i^q = a_i$, or in other words $a_i \in \text{GF}(q)$. Therefore $F(x)$ is a q -polynomial over $\text{GF}(q)$. Q.E.D.

APPENDIX B

FACTORIZATION OF POLYNOMIALS OVER FINITE FIELDS USING DFT OVER FINITE FIELDS

We propose an algorithm for factorization of polynomials over finite fields which can make use of fast Fourier transform (FFT) algorithms for the computation of discrete Fourier transform (DFT). This is essentially a root finding algorithm which computes the roots of a polynomial by DFT methods. This algorithm may be efficiently used if the field in which the roots lie are known and there are no multiple roots. For example, the roots of syndrome polynomials (SPs) discussed in Chapter 6 are known to lie in $GF(2^n)$ and therefore factorization of these polynomials can make use of this algorithm. Thus polynomials representing cyclic codes can be factorized by this method to determine their weight distributions.

We describe this factorization procedure in the following paragraphs:

A polynomial over a finite field $GF(q)$ [q a power of 2] may be expressed in the form

$$\begin{aligned} f(x) &= \sum_{i=0}^n a_i x^i; \text{ where } a_i, x \in GF(q). \\ &= a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n, \end{aligned} \quad (B.1)$$

The polynomial $f(x)$ has a zero at $x = \beta^j$ iff $f(\beta^j) = 0$, where β is a primitive element in $GF(q^m)$, $m \geq 1$. Substituting in (B.1) gives

$$a_0 = \sum_{i=1}^n a_i \beta^{ji} \quad (B.2)$$

Let us assume that $n = q^m - 1$. Then the right hand side (RHS) represents the j^{th} spectral component E_j obtained by taking the finite field DFT of the sequence $a_1 = a_n, a_1,$

..., a_{n-1} where $\beta^j \neq 0$. The DFT coefficients will then be in $GF(q^m)$ since $n|q^m-1$. Note that since $E_n = E_0$, the component a_n occupies the first position in the sequence a_1 . However, n may not always be equal to q^m-1 . But it can be made to, by appending zeroes to a_1 such that its length be equal to q^m-1 . To find the roots of $f(x)$, our first task is to check whether $x = 0$ is a root. This is trivial since if $a_0 = 0$, $f(x) = 0$ at $x = 0$. Further it has 'p' multiple roots at $x = 0$, if the least exponent term in x is x^p . The nonzero roots can be obtained as follows:

The first step would be to see whether the number of terms in x , excluding the constant term a_0 , is equal to q^m-1 . If not, append zeroes to the sequence a_1 such that its length is extended to the nearest $q^{m_1}-1$ where m_1 is the least positive integer which satisfies this relation. Find the DFT of the sequence $a_\xi, a_1, a_2, \dots, a_{\xi-1}$ of length $\xi = q^{m_1}-1$ over $GF(q)$. The DFT coefficients lie in $GF(q^{m_1})$. Search for those spectral components E_j whose value is equal to a_0 . Then those β^j (where β is a primitive element of $GF(q^{m_1})$) form the roots of the polynomial $f(x)$ in the extension field $GF(q^{m_1})$.

If the number of roots so obtained (including $x = 0$) is equal to the degree of the polynomial $f(x)$, then the factorization is complete.

If the nonzero roots of the polynomial $f(x)$ lying in $GF(q^{m_1})$ correspond to 's' conjugacy classes, then $f(x)$ gets factorized into 's' irreducible polynomials over $GF(q)$, each irreducible polynomial being the minimal polynomial of β^{j_i} , where j_i is a member of the conjugacy class modulo $q^{m_1}-1$.

If there are no roots at $x = 0$, and if all the roots so obtained (their number being equal to the degree of the polynomial $f(x)$) in $GF(q^{m_1})$ correspond to a single conjugacy class, then the polynomial $f(x)$ is irreducible over $GF(q)$.

If the number of roots obtained is less than the degree of the polynomial $f(x)$, we have the following two cases:

- (i) The remaining roots lying in an extension field $GF(q^m)$ where $m > m_1$.
- (ii) occurrence of multiple roots.

Consider case (i) first. To search for the remaining roots lying in a further extension field, the sequence length is further extended to $\rho = q^{m_2} - 1$ where $m_2 = m_1 + 1$, and the DFT of the sequence $a_\rho, a_1, a_2, \dots, a_{\rho-1}$ is found. Search for the roots lying in $GF(q^{m_2})$ as before. The procedure is repeated for $m = m_1, m_1 + 1, m_1 + 2, \dots, t$, where t is the degree of the polynomial $f(x)$, or till all the roots are obtained, whichever is earlier.

If the number of roots obtained is still less even at $m = t$, this indicates case (ii), i.e., occurrence of a multiple nonzero root.

We present some examples below to illustrate the above concepts:

Examples

Example B.1: Let $f(x) = 1 + x + x^2 + x^6 + x^7 + x^8 + x^{12} = 1 + f'(x)$ (say).

In this case, $a_0 = 1$, and a_i 's $\in GF(2)$. Since $f'(x)$ has only 12 terms, we can extend it to 15 terms (nearest $2^m - 1$) by adding $0x^{13} + 0x^{14} + 0x^{15}$. Thus let us put $f'(x)$ plus these 3 terms as $f''(x) = 1x + 1x^2 + 0x^3 + 0x^4 + 0x^5 + 1x^6 + 1x^7 + 1x^8 + 0x^9 + 0x^{10} + 0x^{11} + 1x^{12} + 0x^{13} + 0x^{14} + 0x^{15}$.

Taking the coefficients of x as the sequence whose DFT is to be found, in the order mentioned earlier (i.e., with the coefficient of x^{15} first, followed by the coefficients of $x, x^2, x^3, \dots, x^{14}$), the sequence whose DFT is to be found is 0 1 1 0 0 0 1 1 1 0 0 0 1 0 0. The DFT of this sequence exists in $GF(2^4)$. However, it can be verified that none of the DFT coefficients in $GF(2^4)$ is equal to $a_0 (= 1)$.

Hence we proceed to the next extension field $GF(2^5)$. Extend the polynomial $f''(x)$ by adding $0x^{16} + 0x^{17} + \dots + 0x^{31}$. A sequence a_i of length 31 is obtained with the coefficient of x^{31} in the first position followed by the coefficients of $x_i, i = 1, 2, \dots, 30$. Taking the irreducible polynomial for generating $GF(2^5)$ as $x^5 + x^3 + x^2 + x + 1$, the spectral components E_j whose value is $a_0 = 1$, are E_1, E_2, E_4, E_8 and E_{16} . Thus the roots in this field are $\beta, \beta^2, \beta^4, \beta^8$ and β^{16} . It may be noted that they correspond to the same conjugacy class. Thus we get a factor of $f(x)$ which is an irreducible polynomial over $GF(2)$

of degree 5 namely $(x + \beta)(x + \beta^2)(x + \beta^4)(x + \beta^8)(x + \beta^{16}) = x^5 + x^3 + x^2 + x + 1$ [which is the minimal polynomial of β]. Since 7 more roots are to be obtained, we proceed to $GF(2^6)$. It can be verified that there are no roots in $GF(2^6)$. We proceed to $GF(2^7)$ in the same way. Choosing irreducible polynomial for $GF(2^7)$ as $x^7 + x^5 + x^4 + x^3 + 1$, we get the remaining 7 roots in this field as $\beta, \beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32}, \beta^{64}$ which correspond to one conjugacy class and produces another factor of $f(x)$ which is an irreducible polynomial of degree 7 namely $x^7 + x^5 + x^4 + x^3 + 1$. Since all the roots are obtained we do not proceed upto $m = 12$. Thus

$$x^{12} + x^8 + x^7 + x^6 + x^2 + x + 1 = (x^5 + x^3 + x^2 + x + 1)(x^7 + x^5 + x^4 + x^3 + 1).$$

Example B.2:

$$f(x) = x^4 + x^3 + x^2 + x + 1.$$

Number of coefficients of $x = 4$.

The nearest $2^m - 1 = 15$.

The sequence is 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0.

DFT coefficients lie in $GF(2^4)$. Choosing the irreducible polynomial for $GF(2^4)$ as $x^4 + x + 1$, the roots are obtained as $\beta^3, \beta^6, \beta^9$ and β^{12} . All the roots are obtained in this field and they correspond to the same conjugacy class. Hence the above polynomial is irreducible over $GF(2)$.

Example B.3: $f(x) = x^4 + x^2 + 1 = 1 + 0x + 1x^2 + 0x^3 + 1x^4$.

Number of coefficients of $x = 4$.

The nearest $2^m - 1 = 15$.

The sequence whose DFT is to be found is 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0.

Choosing irreducible polynomial for $GF(2^4)$ as $x^4 + x + 1$, the roots are obtained as β^5 and β^{10} . The minimal polynomial of $\beta^5 = x^2 + x + 1$. Since the degree of $f(x)$ is 4, and m is also 4, the procedure is terminated here. Since two more roots are to be obtained still, $f(x)$ has multiple roots at β^5 and β^{10} . Therefore

$$f(x) = (x + \beta^5)(x + \beta^{10})(x + \beta^5)(x + \beta^{10}) = (x^2 + x + 1)^2.$$

Example B.4: $f(x) = \beta^{49}x^4 + \beta^{15}x^2 + \beta^{21}x = \beta^{21}x + \beta^{15}x^2 + 0x^3 + \beta^{49}x^4$, where the coefficients of $f(x)$ are from $GF(2^6)$ and β is a root of the primitive polynomial $x^6 + x + 1$. Since the constant term is 0, and the least exponent of x is 1, the polynomial has a single root at $x = 0$. Here $q = 2^6$. The sequence is extended to the nearest $q^m - 1$. In this case, $m = 1$. Hence the length of the sequence is $2^6 - 1 = 63$. The sequence whose DFT is to be found is $0 \ \beta^{21} \ \beta^{15} \ 0 \ \beta^{49} \ 0 \ 0 \ 0 \dots\dots 0$. The spectral component E_j whose value is $a_0 (= 0$ in this case) is E_{12} only. Hence a second root is β^{12} . Two more roots remain, and hence the sequence is extended to $q^m - 1$ where $m = 2$. The length of the sequence is now $2^{12} - 1 = 4095$. Then the DFT of the sequence (of length 4095 over $GF(2^6)$) is found. The spectral components lie in $GF(2^{12})$. Let γ be a primitive element of $GF(2^{12})$, a root of the primitive polynomial $x^{12} + x^{11} + x^8 + x^6 + 1$. Then $\beta = \gamma^{4095/63} = \gamma^{65}$. Expressing all β^i 's in terms of γ , we have the resulting sequence as $0 \ \gamma^{1365} \ \gamma^{975} \ 0 \ \gamma^{3185} \ 0 \ 0 \ 0 \dots\dots\dots 0$ [4095 data points in this sequence]. On finding the DFT, we see that $GF(2^{12})$ is the splitting field for the above polynomial. Thus all the roots are obtained in this field. Three E_j 's whose value is $a_0 (= 0)$ are E_{780} , E_{2291} and E_{3299} . Hence the roots are γ^{780} [already found in $GF(2^6)$ as β^{12} since $780 = 12 \times 65$], γ^{2291} and γ^{3299} apart from $x = 0$. It may be noted that since the DFT coefficients are in an extension field of $GF(q)$, they satisfy conjugacy constraints [only the last two, since the first namely $\gamma^{780} = \beta^{12} \in GF(2^6)$]:

$(E_j)^q = E_{jq \pmod{q^m - 1}}$. In this case, the indices satisfy the relation $j = 64j \pmod{4095}$. Thus $\gamma^{(2291 \times 64) \pmod{4095}} = \gamma^{3299}$. The procedure is terminated since all the four roots are obtained.

APPENDIX C

TABLES OF FINITE FIELDS

Table C.1: $GF(2^2)$

Minimal Polynomial: $x^2 + x + 1$

Standard Basis = $\{\alpha^0, \alpha\}$

$$\alpha^j = m_{j1} \alpha + m_{j0} \alpha^0$$

Normal Basis = $\{\alpha, \alpha^2\}$

$$\beta^j = m_{j1} \alpha^2 + m_{j0} \alpha$$

j	$m_{j1} m_{j0}$	
$-\infty$	0	0
0	0	1
1	1	0
2	1	1

j	$m_{j1} m_{j0}$	
$-\infty$	0	0
0	1	1
1	0	1
2	1	0

Table C.2: $GF(2^3)$

Minimal Polynomial = $x^3 + x^2 + 1$

Standard Basis = $\{\alpha^0, \alpha, \alpha^2\}$

$$\alpha^j = m_{j2} \alpha^2 + m_{j1} \alpha + m_{j0} \alpha^0$$

Normal Basis = $\{\alpha, \alpha^2, \alpha^4\}$

$$\beta^j = m_{j2} \alpha^4 + m_{j1} \alpha^2 + m_{j0} \alpha$$

j	$m_{j2} m_{j1} m_{j0}$		
$-\infty$	0	0	0
0	0	0	1
1	0	1	0
2	1	0	0
3	1	0	1
4	1	1	1
5	0	1	1
6	1	1	0

j	$m_{j2} m_{j1} m_{j0}$		
$-\infty$	0	0	0
0	1	1	1
1	0	0	1
2	0	1	0
3	1	0	1
4	1	0	0
5	1	1	0
6	0	1	1

Table C.3: $GF(2^4)$ Minimal Polynomial: $x^4 + x + 1$ Standard Basis = $\{\alpha^0, \alpha, \alpha^2, \alpha^3\}$ Normal Basis = $\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\}$

$$\alpha^j = m_{j3}\alpha^3 + m_{j2}\alpha^2 + m_{j1}\alpha + m_{j0}\alpha^0$$

$$\alpha^j = m_{j3}\alpha^9 + m_{j2}\alpha^{12} + m_{j1}\alpha^6 + m_{j0}\alpha^3$$

j	m_{j3}	m_{j2}	m_{j1}	m_{j0}
$-\infty$	0	0	0	0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	1	1
5	0	1	1	0
6	1	1	0	0
7	1	0	1	1
8	0	1	0	1
9	1	0	1	0
10	0	1	1	1
11	1	1	1	0
12	1	1	1	1
13	1	1	0	1
14	1	0	0	1

j	m_{j3}	m_{j2}	m_{j1}	m_{j0}
$-\infty$	0	0	0	0
0	1	1	1	1
1	1	0	0	1
2	0	0	1	1
3	0	0	0	1
4	0	1	1	0
5	1	0	1	0
6	0	0	1	0
7	0	1	1	1
8	1	1	0	0
9	1	0	0	0
10	0	1	0	1
11	1	0	1	1
12	0	1	0	0
13	1	1	0	1
14	1	1	1	0

Table C.4: $GF(2^5)$ Minimal Polynomial: $x^5 + x^4 + x^3 + x^2 + 1$ Standard Basis = $\{\alpha^0, \alpha, \alpha^2, \alpha^3, \alpha^4\}$

$$\alpha^j = m_{j4}\alpha^4 + m_{j3}\alpha^3 + m_{j2}\alpha^2 + m_{j1}\alpha + m_{j0}\alpha^0$$

Normal Basis = $\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$

$$\vartheta^j = m_{j4}\alpha^{16} + m_{j3}\alpha^8 + m_{j2}\alpha^4 + m_{j1}\alpha^2 + m_{j0}\alpha$$

j	m_{j4}	m_{j3}	m_{j2}	m_{j1}	m_{j0}
$-\infty$	0	0	0	0	0
0	0	0	0	0	1
1	0	0	0	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	1	0	0	0	0
5	1	1	1	0	1
6	0	0	1	1	1
7	0	1	1	1	0
8	1	1	1	0	0
9	0	0	1	0	1
10	0	1	0	1	0
11	1	0	1	0	0
12	1	0	1	0	1
13	1	0	1	1	1
14	1	0	0	1	1
15	1	1	0	1	1
16	0	1	0	1	1
17	1	0	1	1	0
18	1	0	0	0	1
19	1	1	1	1	1
20	0	0	0	1	1
21	0	0	1	1	0
22	0	1	1	0	0
23	1	1	0	0	0
24	0	1	1	0	1
25	1	1	0	1	0
26	0	1	0	0	1
27	1	0	0	1	0
28	1	1	0	0	1
29	0	1	1	1	1
30	1	1	1	1	0

j	m_{j4}	m_{j3}	m_{j2}	m_{j1}	m_{j0}
$-\infty$	0	0	0	0	0
0	1	1	1	1	1
1	0	0	0	0	1
2	0	0	0	1	0
3	0	1	1	1	0
4	0	0	1	0	0
5	1	0	1	1	1
6	1	1	1	0	0
7	0	1	1	0	1
8	0	1	0	0	0
9	1	1	1	0	1
10	0	1	1	1	1
11	0	0	1	1	0
12	1	1	0	0	1
13	1	1	0	0	0
14	1	1	0	1	0
15	1	0	1	0	0
16	1	0	0	0	0
17	0	0	1	1	1
18	1	1	0	1	1
19	1	0	1	1	0
20	1	1	1	1	0
21	0	0	0	1	1
22	0	1	1	0	0
23	0	1	0	1	0
24	1	0	0	1	1
25	0	1	0	1	1
26	1	0	0	0	1
27	0	0	1	0	1
28	1	0	1	0	1
29	1	0	0	1	0
30	0	1	0	0	1

REFERENCES

1. Ninomiya, I., 'A Theory of Coordinate Representation of Switching Functions', *Memoirs. Fac. Engg. Nagoya Univ.*, vol.10, 1958, pp 175-190.
2. Ninomiya, I., 'A Study of the Structures of Boolean Functions and its Applications to Synthesis of Switching Circuits', *Memoirs. Fac. Engg. Nagoya Univ.*, vol.13, 1961, pp 149-363.
3. Bartee, T.C. and Schneider, D.I., 'Computation with Finite Fields', *Inform. Contr.*, vol.6, 1963, pp 79-98.
4. Benjauthrit, B. and Reed, I.S., 'Galois Switching Functions and their Applications', *IEEE Trans. Comput.*, vol. C-25, 1976, pp 78-86.
5. Benjauthrit, B. and Reed, I.S., 'On the Fundamental Structure of Galois Switching Functions', *IEEE Trans. Comput.*, vol. C-27, No.8, 1978, pp 757-762.
6. Menger, K.S., Jr., 'A Transform for Logic Networks', *IEEE Trans. Comput.*, vol. C-18, 1969, pp 241-250.
7. Pradhan, D.K. and Patel, A.M., 'Reed-Muller like Canonic Forms for Multivalued Functions', *IEEE Trans. Comput. (Corresp.)*, 1975, pp 206-210.
8. Mukhopadhyay, A. and Schmitz, G., 'Minimization of Exclusive-or and Logical Equivalence Switching Circuits', *IEEE Trans. Comput.*, vol. C-19, 1970, pp 132-140.
9. Pradhan, D.K., 'A Theory of Galois Switching Functions', *IEEE Trans. Comput.*, vol. C-27, 1978, pp 239-248.
10. Takahashi, I., 'Switching Functions Constructed by Galois Extension Fields', *Inform. Contr.*, vol.48, No.2, 1981, pp 95-108.

11. Davio, M., Deschamps, J.P. and Thayse, A., *Discrete and Switching Functions*, Georgi Publishing Co. and McGraw Hill, Switzerland, 1978.
12. Siddiqi, M.U. and Sinha, V.P., 'Signals and Systems over Finite Groups and Monoids', *Proc. Indo-US Workshop on Systems and Signal Processing*, I.I.Sc., Bangalore, Jan, 1988.
13. Hurst, S.L., *The Logical Processing of Digital Signals*, Edward Arnold, London, 1978.
14. Hurst, S.L., Miller, D.M. and Muzio, J.C., *Spectral Techniques in Digital Logic*, Academic Press, 1985.
15. Karpovsky, M.G., *Finite Orthogonal Series in the Design of Digital Devices*, Halstead Press, 1976.
16. Mukhopadhyay, A. (Ed.), *Recent Developments in Switching Theory*, Academic Press, New York, 1971.
17. Edwards, C.R., 'The Application of Rademacher-Walsh Transform to Boolean Function Classification and Threshold Logic Synthesis', *IEEE Trans. Comput.*, vol. C-24, No.1, 1975, pp 48-62.
18. Mattson, H.F., Jr., and Solomon, G., 'A New Treatment of Bose-Chaudhuri Codes', *J. Soc. Indust. Appl. Math.*, vol. 9, 1961, pp 654-669.
19. MacWilliams, F.J. and Sloane, N.J.A., *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.
20. Blahut, R.E., *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.
21. Blahut, R.E., 'Algebraic Codes in the Frequency Domain', *CISM Courses and Lectures*, No. 258, Springer-Verlag, New York.
22. Kasami, T., Lin, S. and Peterson, W.W., 'New Generalizations of the Reed-Muller codes, Part-I : Primitive Codes', *IEEE Trans. Inform. Theory*, vol. IT-14, No.2, 1968, pp 189-205.

23. Ore, O., 'Theory of Non Commutative Polynomials', *Ann. of Math.*, vol.34, 1933, pp 480–508.
24. Ore, O., 'On a Special Class of Polynomials', *Trans. Amer. Math. Soc.*, vol.35, 1933, pp 559–584.
25. Ore, O., 'Contributions to the Theory of Finite Fields', *Trans. Amer. Math. Soc.*, vol. 36, 1934, pp 243–274.
26. Ore, O., 'Some Studies on Cyclic Determinants', *Duke Math. J.*, vol. 18, 1951, pp 343–354.
27. Lidl, R. and Niederreiter, H., *Finite Fields*, Encyclopedia of Mathematics and its applications, vol.20, Cambridge University Press, Cambridge, 1983.
28. Berlekamp, E.R., *Algebraic Coding Theory*, McGraw–Hill, New York, 1968.
29. Peterson, W.W. and Weldon, E.J., Jr., *Error–Correcting Codes*, 2nd ed., M.I.T. Press, Cambridge, Mass., 1972.
30. Jamison, R.E., 'Covering Finite Fields with Cosets of Subspaces', *J. Combinatorial Theory Ser. A*, vol. 22, 1977, pp 253–266.
31. Siddiqi, M.U., *A Study of Permutation–Invariant Linear Systems*, Ph.D. Thesis, Indian Institute of Technology, Kanpur, India, 1976.
32. Madhusudhana, H.S., *On Abelian Codes Which are closed under Cyclic Shifts*, M.Tech. Thesis, Indian Institute of Technology, Kanpur, India, 1986.
33. Blake, I.F. and Mullin, R.C., *An Introduction to Algebraic and Combinatorial Coding Theory*, Academic Press, New York, 1976.
34. Birkhoff, G. and Bartee, T.C., *Modern Applied Algebra*, McGraw Hill, New York, 1970.
35. Herstein, I.N., *Topics in Algebra*, Vikas Publishing House, New Delhi, 1976.
36. Zariski, O. and Samuel, P., *Commutative Algebra, Vol.1*, Van Nostrand, Princeton NJ, 1958.

AJJ40

Th
6.

EE-1890-D-VAR-GAL